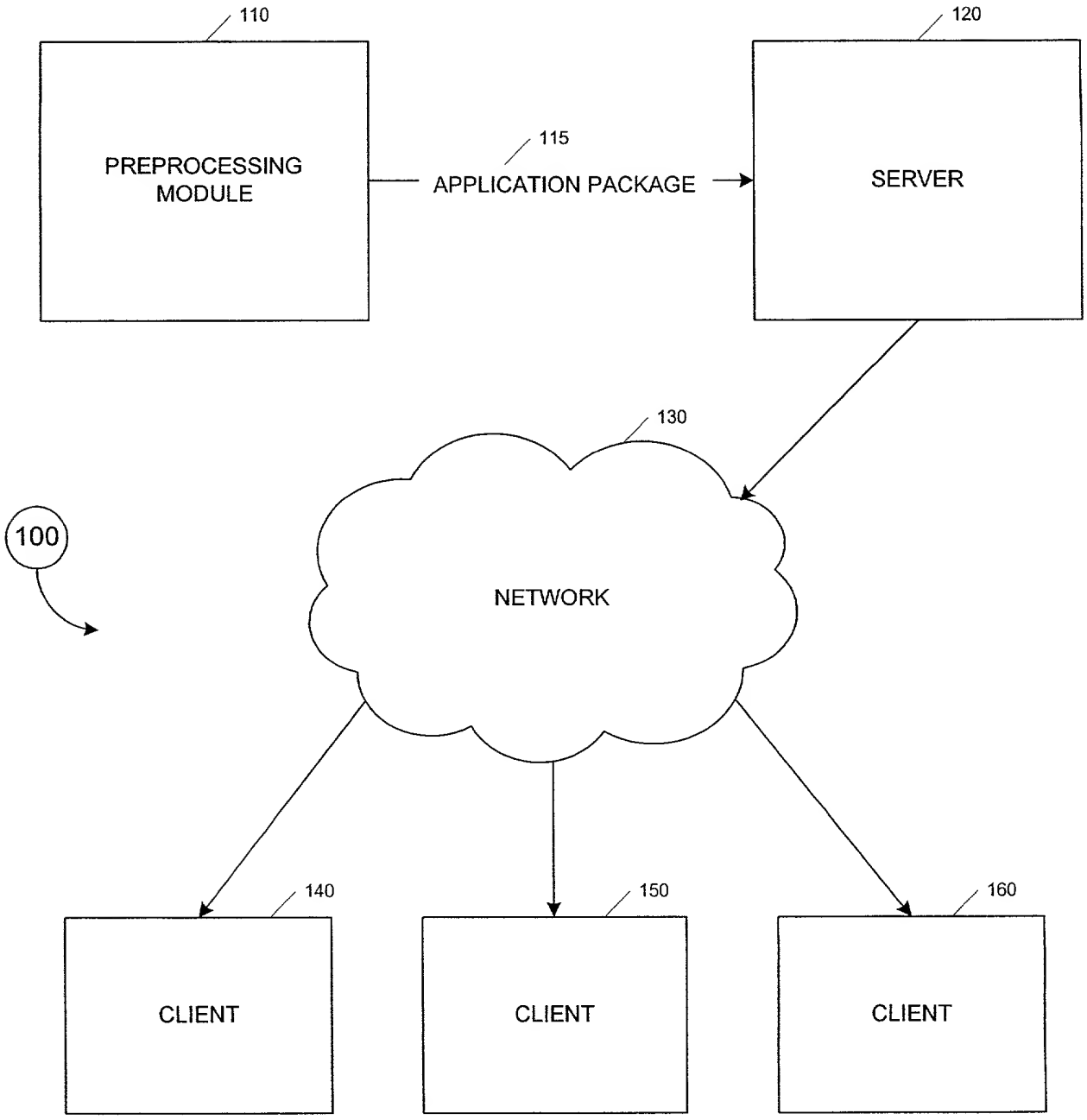
110
PREPROCESSING MODULE
115
APPLICATION PACKAGE
120
SERVER
130
NETWORK
100
140
CLIENT
150
CLIENT
160
CLIENT

FIG. 1

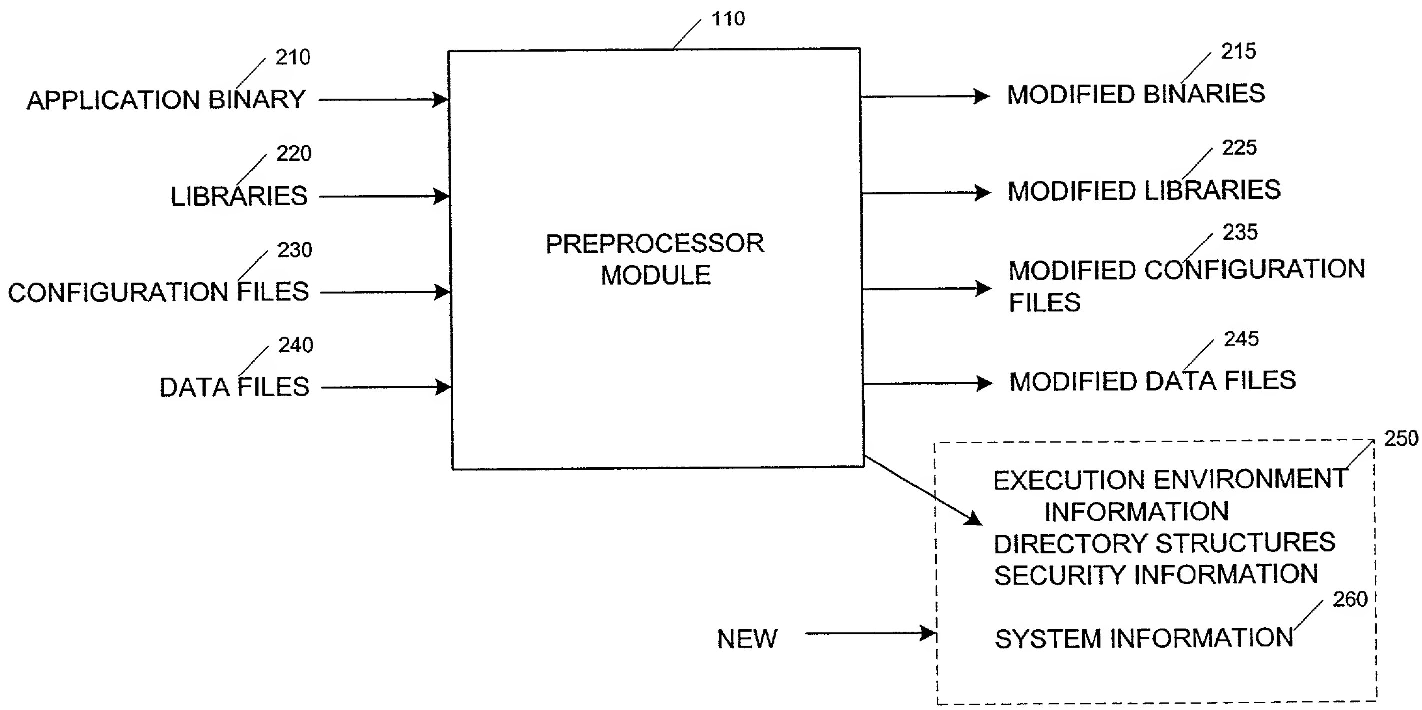
110
210
APPLICATION BINARY
220
LIBRARIES
230
CONFIGURATION FILES
240
DATA FILES
PREPROCESSOR MODULE
215
MODIFIED BINARIES
225
MODIFIED LIBRARIES
235
MODIFIED CONFIGURATION FILES
245
MODIFIED DATA FILES
250
EXECUTION ENVIRONMENT INFORMATION
DIRECTORY STRUCTURES
SECURITY INFORMATION
260
SYSTEM INFORMATION
NEW

FIG. 2

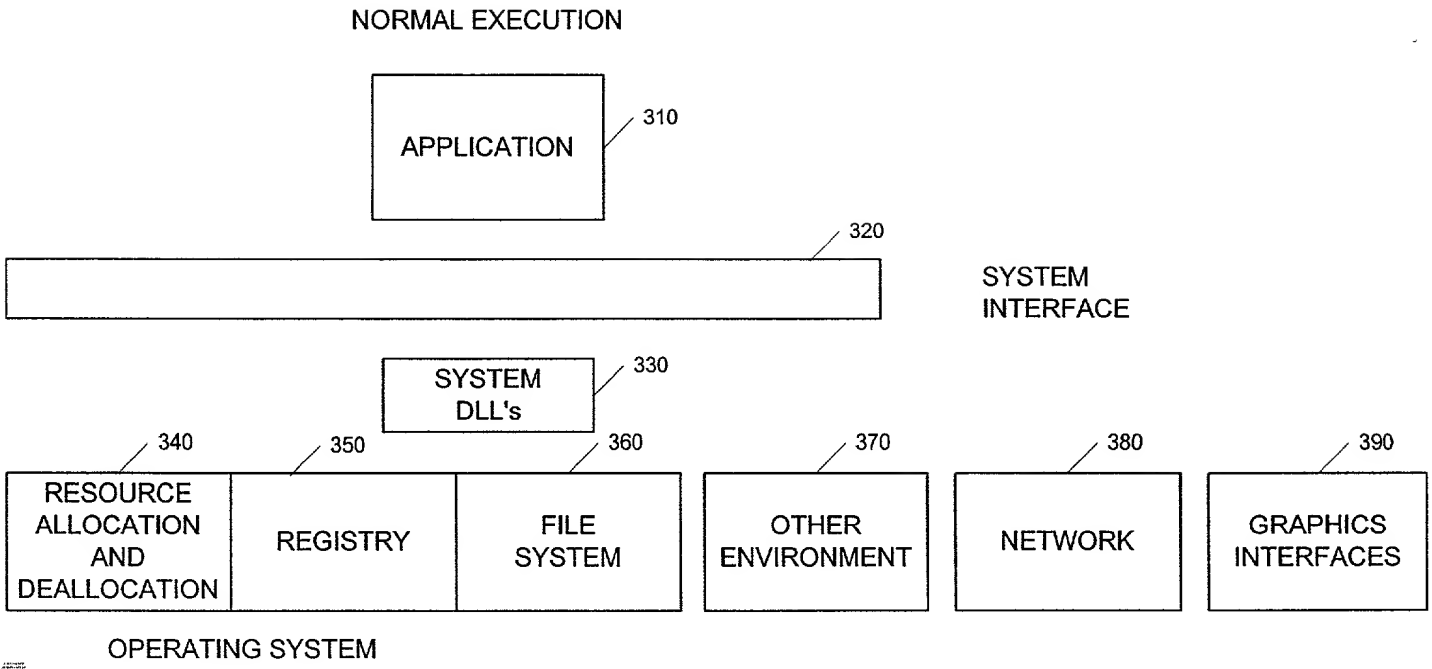
NORMAL EXECUTION
APPLICATION
310
320
SYSTEM INTERFACE
SYSTEM DLL's
330
340
350
360
370
380
390
RESOURCE ALLOCATION AND DEALLOCATION
REGISTRY
FILE SYSTEM
OTHER ENVIRONMENT
NETWORK
GRAPHICS INTERFACES
OPERATING SYSTEM

FIG. 3

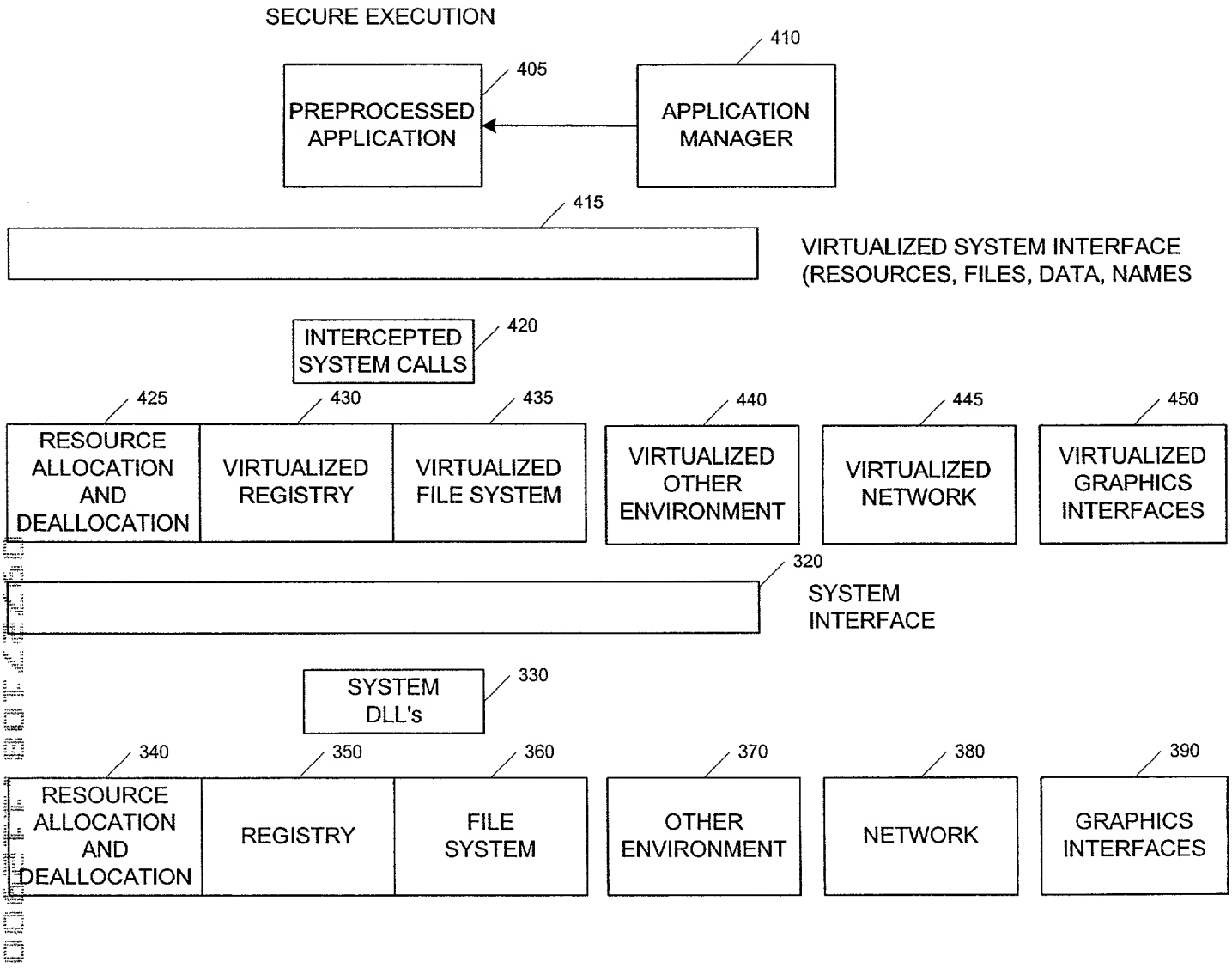
SECURE EXECUTION
410
405
PREPROCESSED APPLICATION
APPLICATION MANAGER
415
VIRTUALIZED SYSTEM INTERFACE (RESOURCES, FILES, DATA, NAMES
INTERCEPTED SYSTEM CALLS
420
425
430
435
440
445
450
RESOURCE ALLOCATION AND DEALLOCATION
VIRTUALIZED REGISTRY
VIRTUALIZED FILE SYSTEM
VIRTUALIZED OTHER ENVIRONMENT
VIRTUALIZED NETWORK
VIRTUALIZED GRAPHICS INTERFACES
320
SYSTEM INTERFACE
SYSTEM DLL's
330
340
350
360
370
380
390
RESOURCE ALLOCATION AND DEALLOCATION
REGISTRY
FILE SYSTEM
OTHER ENVIRONMENT
NETWORK
GRAPHICS INTERFACES

FIG. 4

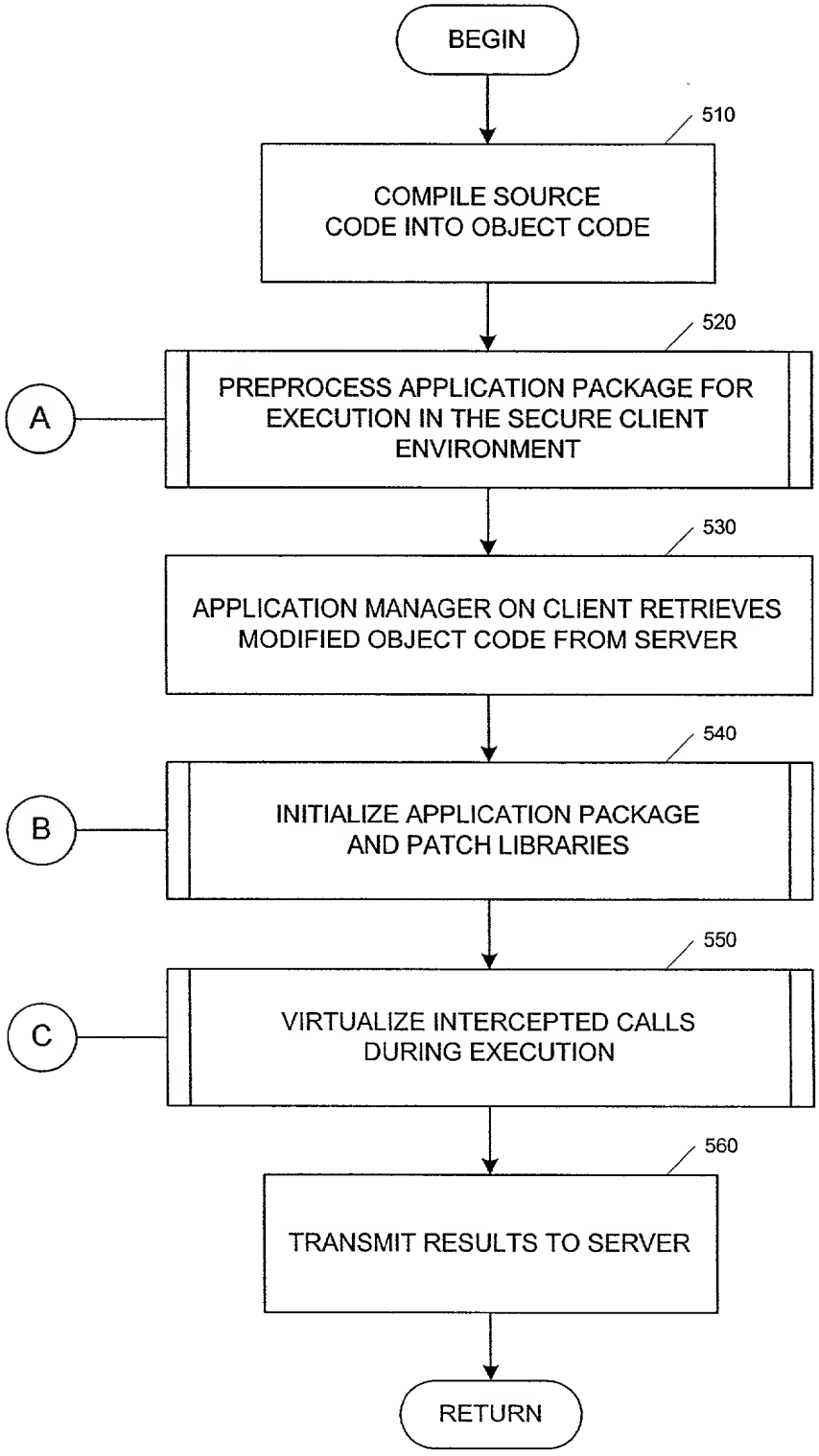
BEGIN
510
COMPILE SOURCE CODE INTO OBJECT CODE
520
A
PREPROCESS APPLICATION PACKAGE FOR EXECUTION IN THE SECURE CLIENT ENVIRONMENT
530
APPLICATION MANAGER ON CLIENT RETRIEVES MODIFIED OBJECT CODE FROM SERVER
540
B
INITIALIZE APPLICATION PACKAGE AND PATCH LIBRARIES
550
C
VIRTUALIZE INTERCEPTED CALLS DURING EXECUTION
560
TRANSMIT RESULTS TO SERVER
RETURN

FIG. 5

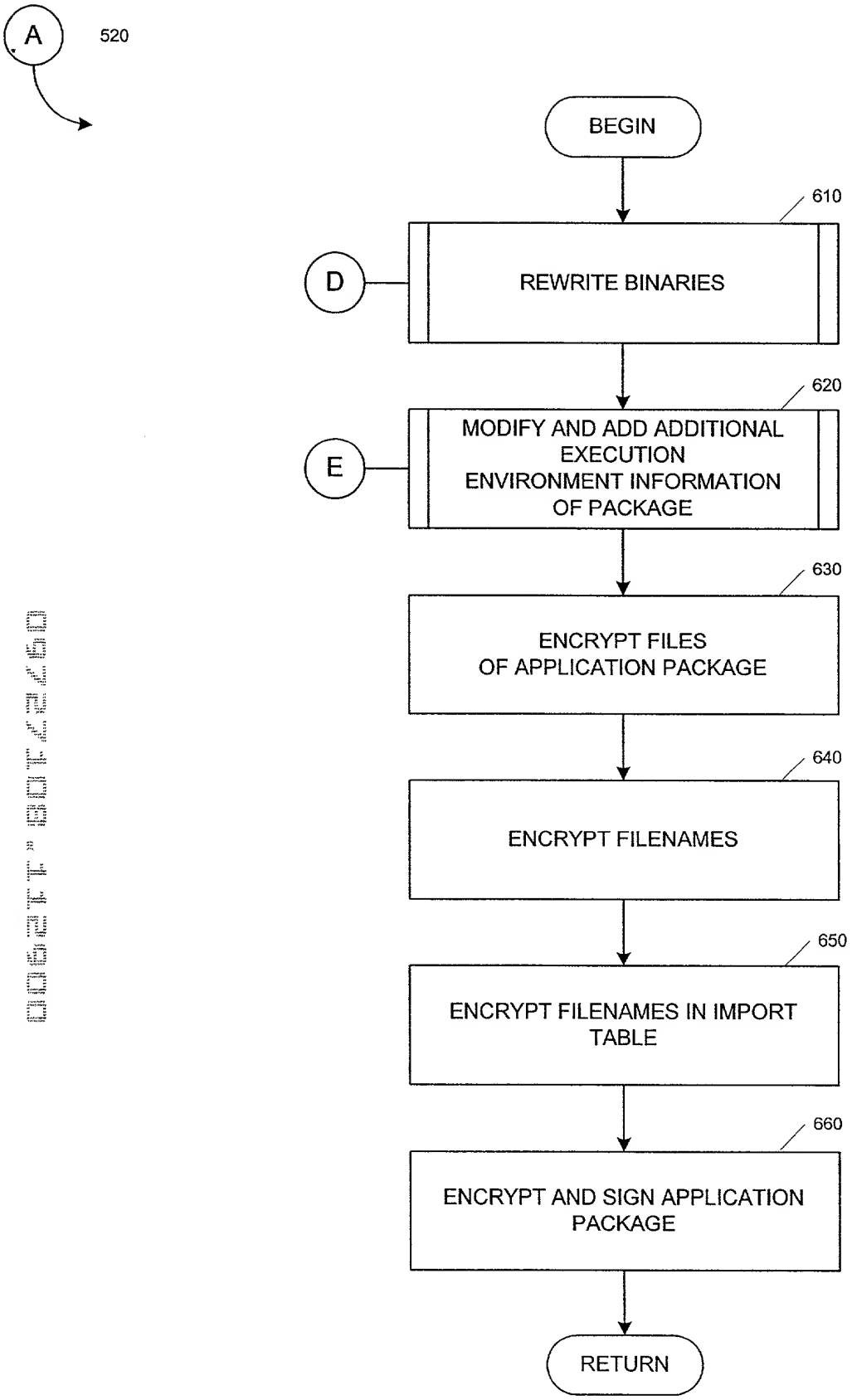
A
520
BEGIN
610
D
REWRITE BINARIES
620
E
MODIFY AND ADD ADDITIONAL EXECUTION ENVIRONMENT INFORMATION OF PACKAGE
630
ENCRYPT FILES OF APPLICATION PACKAGE
640
ENCRYPT FILENAMES
650
ENCRYPT FILENAMES IN IMPORT TABLE
660
ENCRYPT AND SIGN APPLICATION PACKAGE
RETURN

FIG. 6

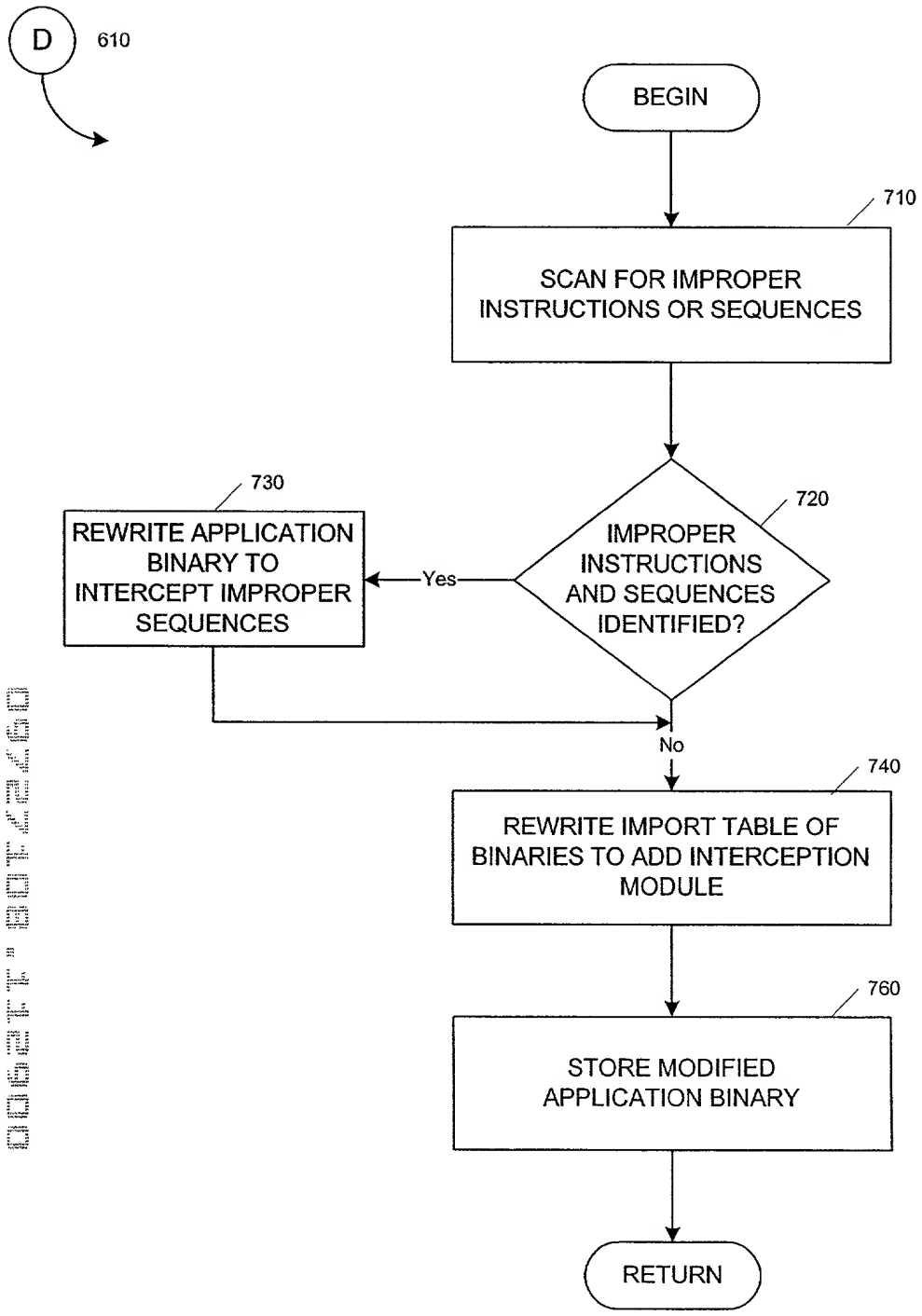
D
610
BEGIN
710
SCAN FOR IMPROPER INSTRUCTIONS OR SEQUENCES
720
IMPROPER INSTRUCTIONS AND SEQUENCES IDENTIFIED?
Yes
730
REWRITE APPLICATION BINARY TO INTERCEPT IMPROPER SEQUENCES
No
740
REWRITE IMPORT TABLE OF BINARIES TO ADD INTERCEPTION MODULE
760
STORE MODIFIED APPLICATION BINARY
RETURN

FIG. 7

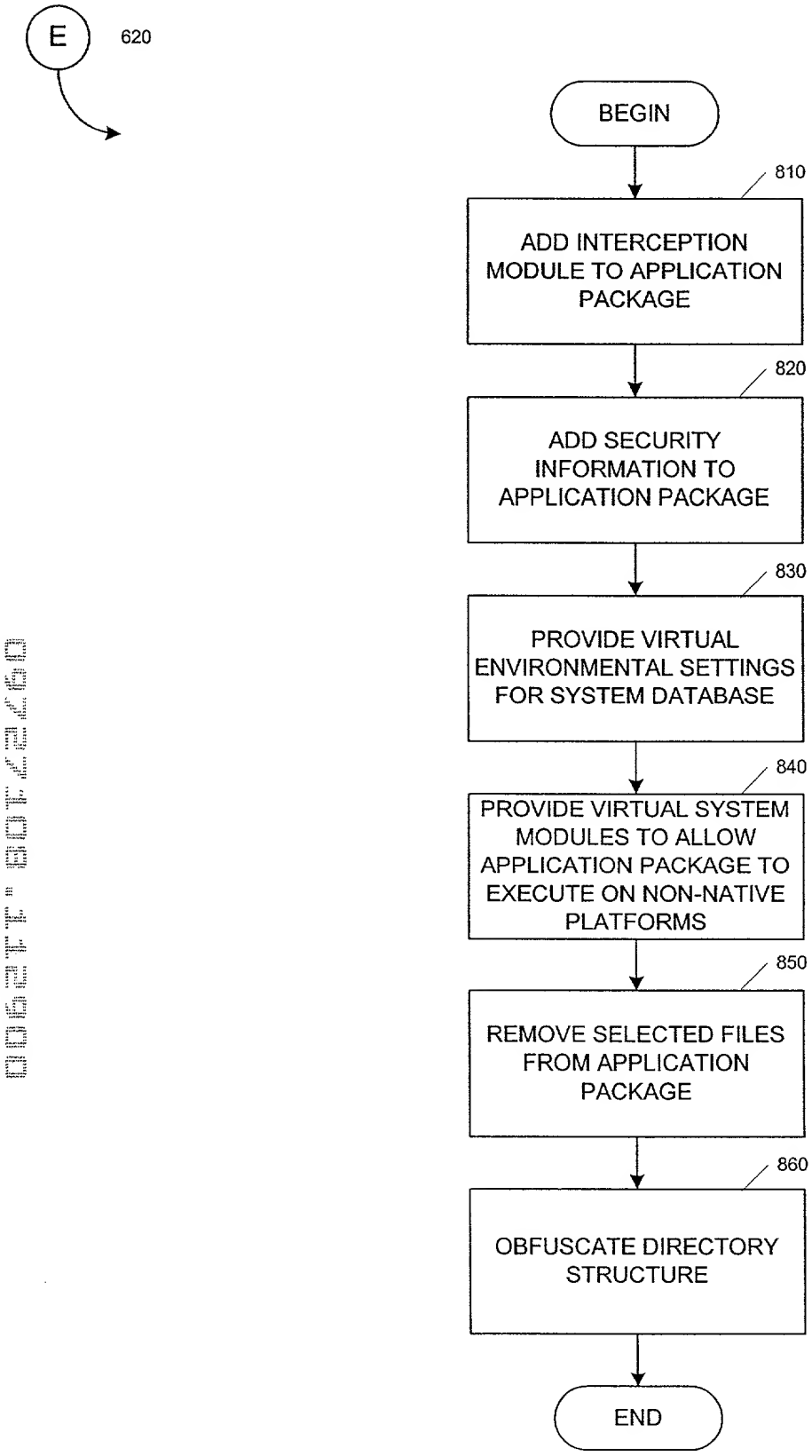
E
620
BEGIN
810
ADD INTERCEPTION MODULE TO APPLICATION PACKAGE
820
ADD SECURITY INFORMATION TO APPLICATION PACKAGE
830
PROVIDE VIRTUAL ENVIRONMENTAL SETTINGS FOR SYSTEM DATABASE
840
PROVIDE VIRTUAL SYSTEM MODULES TO ALLOW APPLICATION PACKAGE TO EXECUTE ON NON-NATIVE PLATFORMS
850
REMOVE SELECTED FILES FROM APPLICATION PACKAGE
860
OBFUSCATE DIRECTORY STRUCTURE
END

FIG. 8

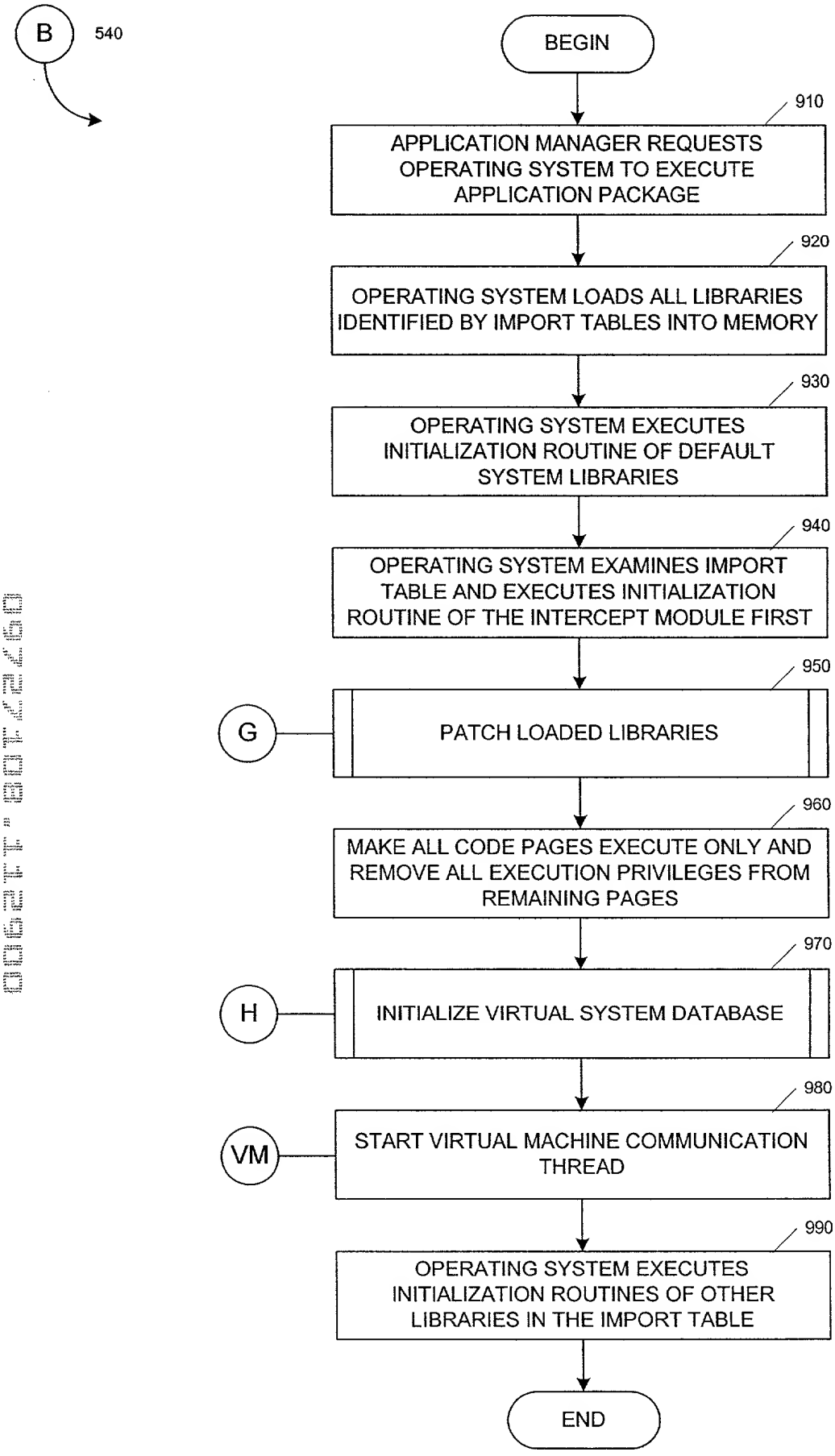
B
540
BEGIN
910
APPLICATION MANAGER REQUESTS OPERATING SYSTEM TO EXECUTE APPLICATION PACKAGE
920
OPERATING SYSTEM LOADS ALL LIBRARIES IDENTIFIED BY IMPORT TABLES INTO MEMORY
930
OPERATING SYSTEM EXECUTES INITIALIZATION ROUTINE OF DEFAULT SYSTEM LIBRARIES
940
OPERATING SYSTEM EXAMINES IMPORT TABLE AND EXECUTES INITIALIZATION ROUTINE OF THE INTERCEPT MODULE FIRST
950
G
PATCH LOADED LIBRARIES
960
MAKE ALL CODE PAGES EXECUTE ONLY AND REMOVE ALL EXECUTION PRIVILEGES FROM REMAINING PAGES
970
H
INITIALIZE VIRTUAL SYSTEM DATABASE
980
VM
START VIRTUAL MACHINE COMMUNICATION THREAD
990
OPERATING SYSTEM EXECUTES INITIALIZATION ROUTINES OF OTHER LIBRARIES IN THE IMPORT TABLE
END

FIG. 9

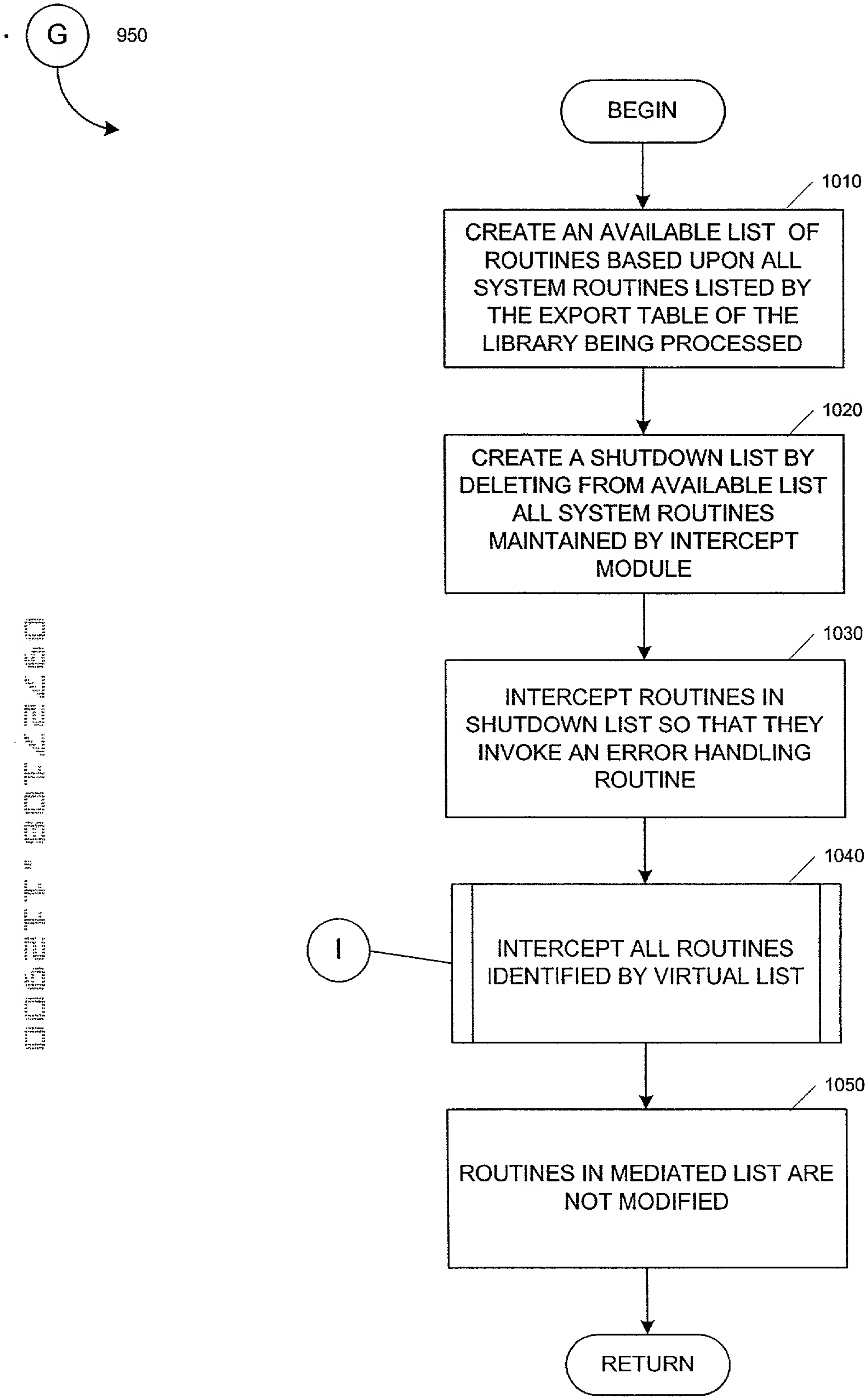
G
950
BEGIN
1010
CREATE AN AVAILABLE LIST OF ROUTINES BASED UPON ALL SYSTEM ROUTINES LISTED BY THE EXPORT TABLE OF THE LIBRARY BEING PROCESSED
1020
CREATE A SHUTDOWN LIST BY DELETING FROM AVAILABLE LIST ALL SYSTEM ROUTINES MAINTAINED BY INTERCEPT MODULE
1030
INTERCEPT ROUTINES IN SHUTDOWN LIST SO THAT THEY INVOKE AN ERROR HANDLING ROUTINE
1040
I
INTERCEPT ALL ROUTINES IDENTIFIED BY VIRTUAL LIST
1050
ROUTINES IN MEDIATED LIST ARE NOT MODIFIED
RETURN

FIG. 10

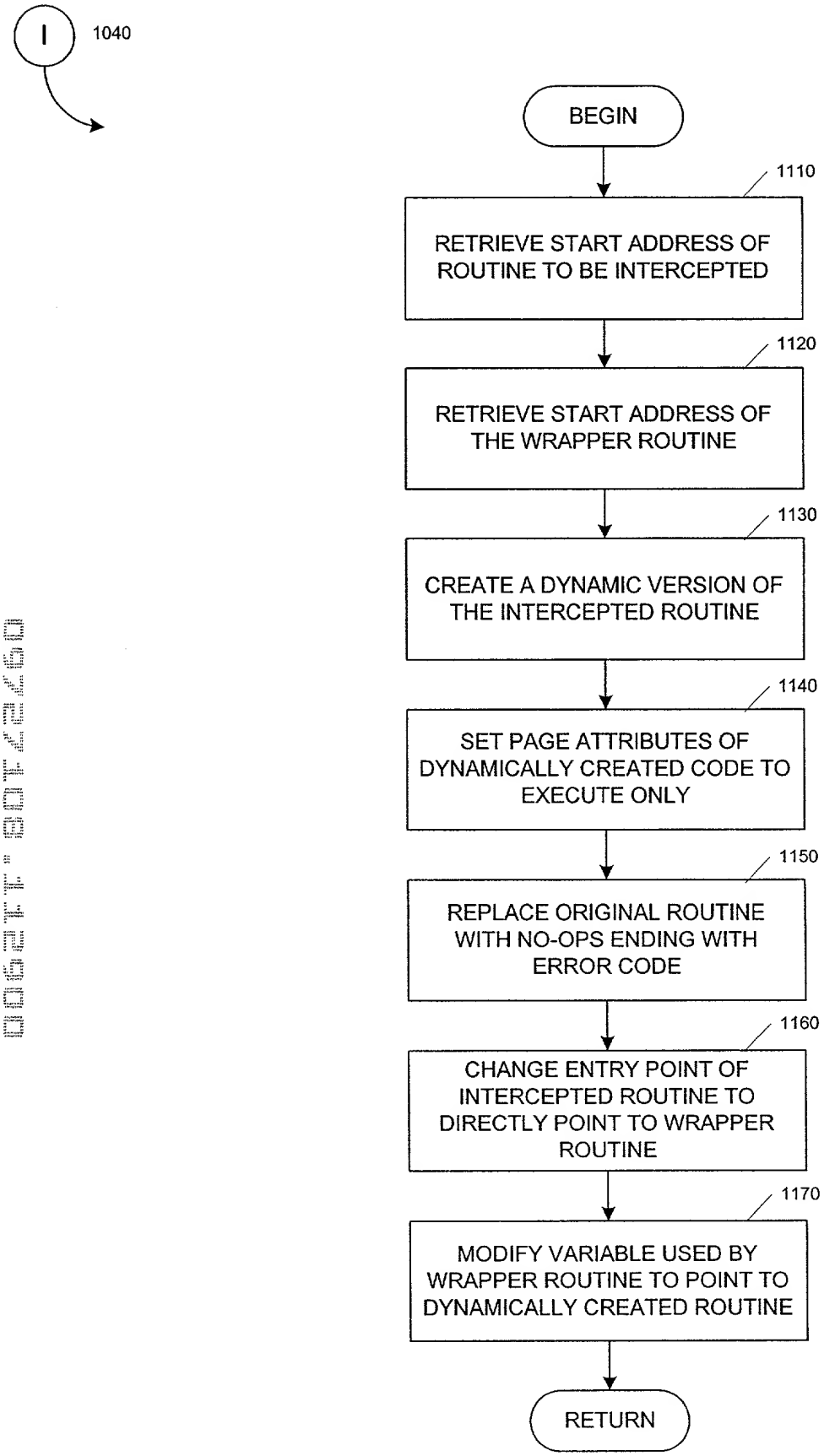
I
1040
BEGIN
1110
RETRIEVE START ADDRESS OF ROUTINE TO BE INTERCEPTED
1120
RETRIEVE START ADDRESS OF THE WRAPPER ROUTINE
1130
CREATE A DYNAMIC VERSION OF THE INTERCEPTED ROUTINE
1140
SET PAGE ATTRIBUTES OF DYNAMICALLY CREATED CODE TO EXECUTE ONLY
1150
REPLACE ORIGINAL ROUTINE WITH NO-OPS ENDING WITH ERROR CODE
1160
CHANGE ENTRY POINT OF INTERCEPTED ROUTINE TO DIRECTLY POINT TO WRAPPER ROUTINE
1170
MODIFY VARIABLE USED BY WRAPPER ROUTINE TO POINT TO DYNAMICALLY CREATED ROUTINE
RETURN

FIG. 11

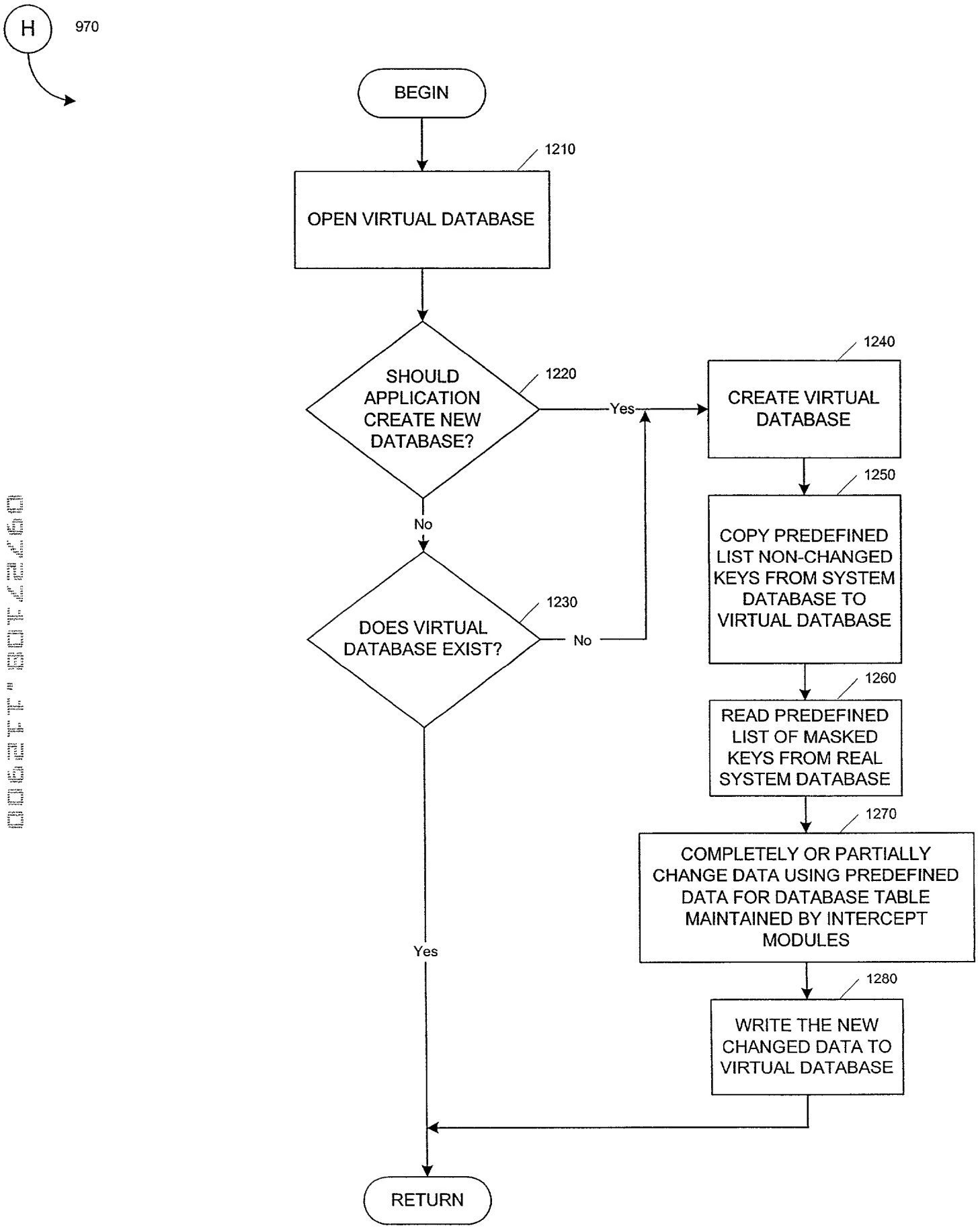
H
970
BEGIN
1210
OPEN VIRTUAL DATABASE
SHOULD APPLICATION CREATE NEW DATABASE?
1220
Yes
No
1240
CREATE VIRTUAL DATABASE
1250
COPY PREDEFINED LIST NON-CHANGED KEYS FROM SYSTEM DATABASE TO VIRTUAL DATABASE
DOES VIRTUAL DATABASE EXIST?
1230
No
Yes
1260
READ PREDEFINED LIST OF MASKED KEYS FROM REAL SYSTEM DATABASE
1270
COMPLETELY OR PARTIALLY CHANGE DATA USING PREDEFINED DATA FOR DATABASE TABLE MAINTAINED BY INTERCEPT MODULES
1280
WRITE THE NEW CHANGED DATA TO VIRTUAL DATABASE
RETURN

FIG. 12

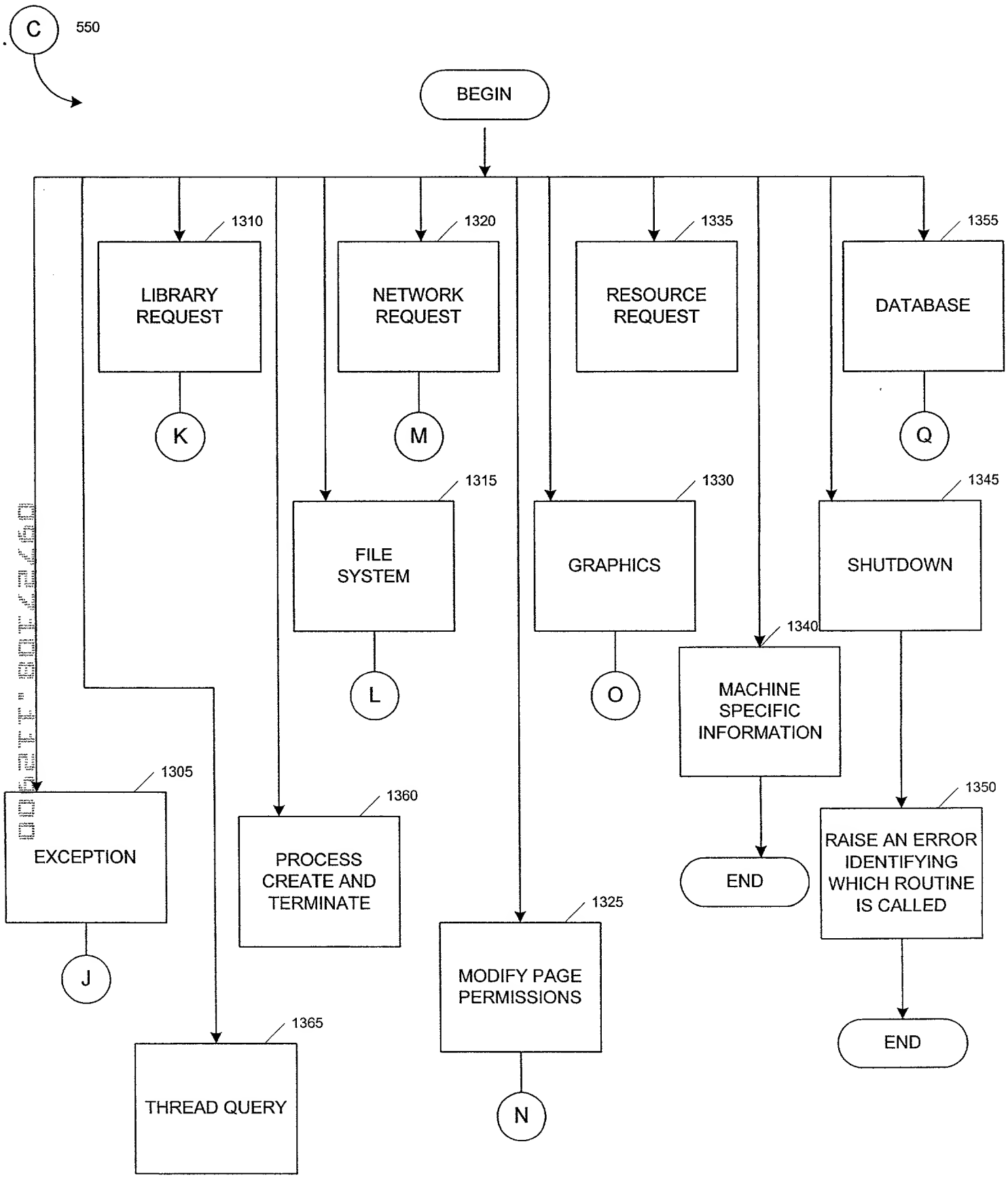
C
550
BEGIN
1310
LIBRARY REQUEST
1320
NETWORK REQUEST
1335
RESOURCE REQUEST
1355
DATABASE
K
M
Q
1315
FILE SYSTEM
1330
GRAPHICS
1345
SHUTDOWN
L
O
1340
MACHINE SPECIFIC INFORMATION
1305
EXCEPTION
1360
PROCESS CREATE AND TERMINATE
1350
RAISE AN ERROR IDENTIFYING WHICH ROUTINE IS CALLED
END
J
1325
MODIFY PAGE PERMISSIONS
1365
THREAD QUERY
N
END

FIG. 13

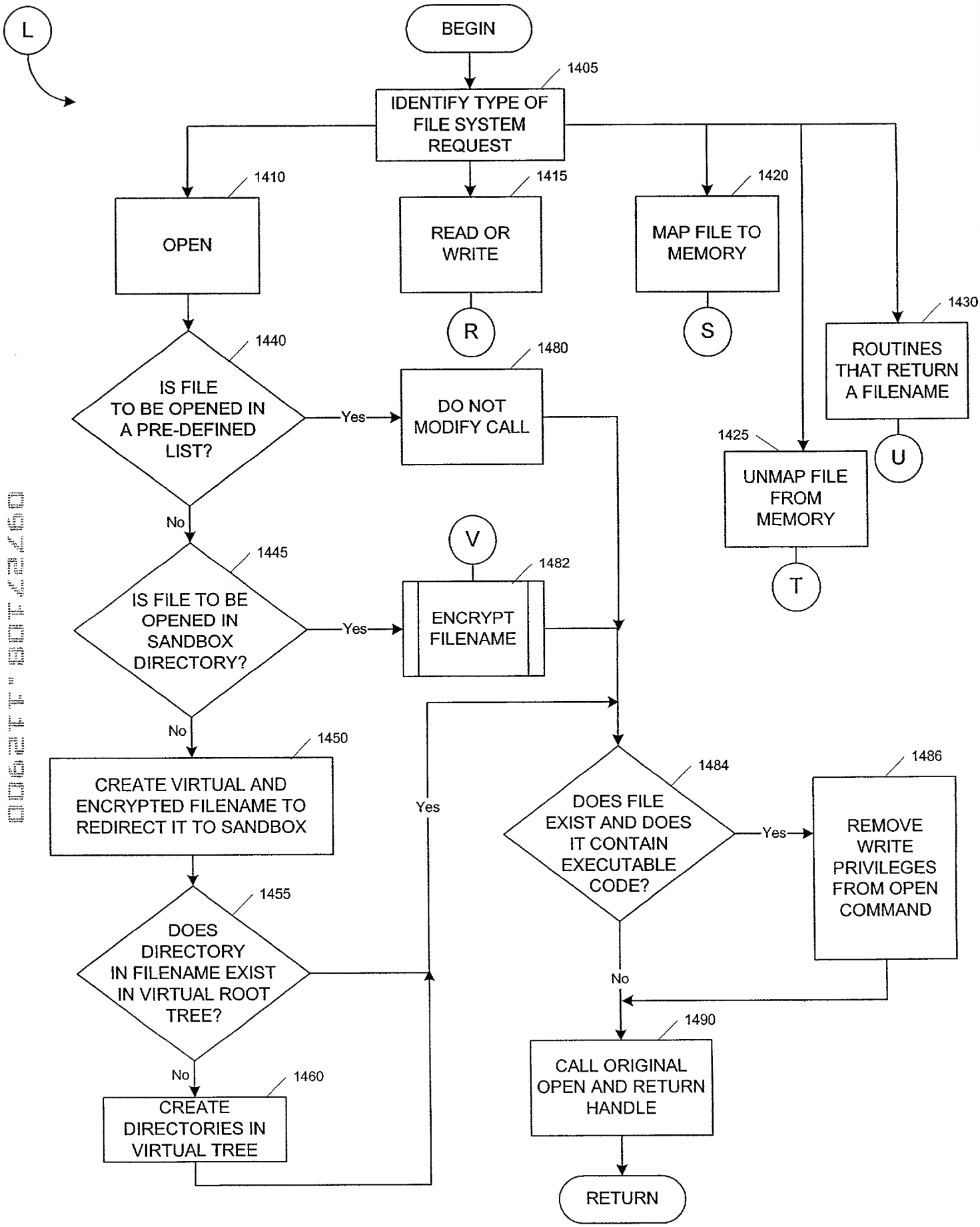
L
BEGIN
1405
IDENTIFY TYPE OF FILE SYSTEM REQUEST
1410
OPEN
1415
READ OR WRITE
1420
MAP FILE TO MEMORY
1430
ROUTINES THAT RETURN A FILENAME
R
S
U
1440
IS FILE TO BE OPENED IN A PRE-DEFINED LIST?
Yes
1480
DO NOT MODIFY CALL
1425
UNMAP FILE FROM MEMORY
T
No
V
1445
IS FILE TO BE OPENED IN SANDBOX DIRECTORY?
Yes
1482
ENCRYPT FILENAME
No
1450
CREATE VIRTUAL AND ENCRYPTED FILENAME TO REDIRECT IT TO SANDBOX
Yes
1484
DOES FILE EXIST AND DOES IT CONTAIN EXECUTABLE CODE?
Yes
1486
REMOVE WRITE PRIVILEGES FROM OPEN COMMAND
1455
DOES DIRECTORY IN FILENAME EXIST IN VIRTUAL ROOT TREE?
No
1490
CALL ORIGINAL OPEN AND RETURN HANDLE
No
1460
CREATE DIRECTORIES IN VIRTUAL TREE
RETURN

FIG. 14

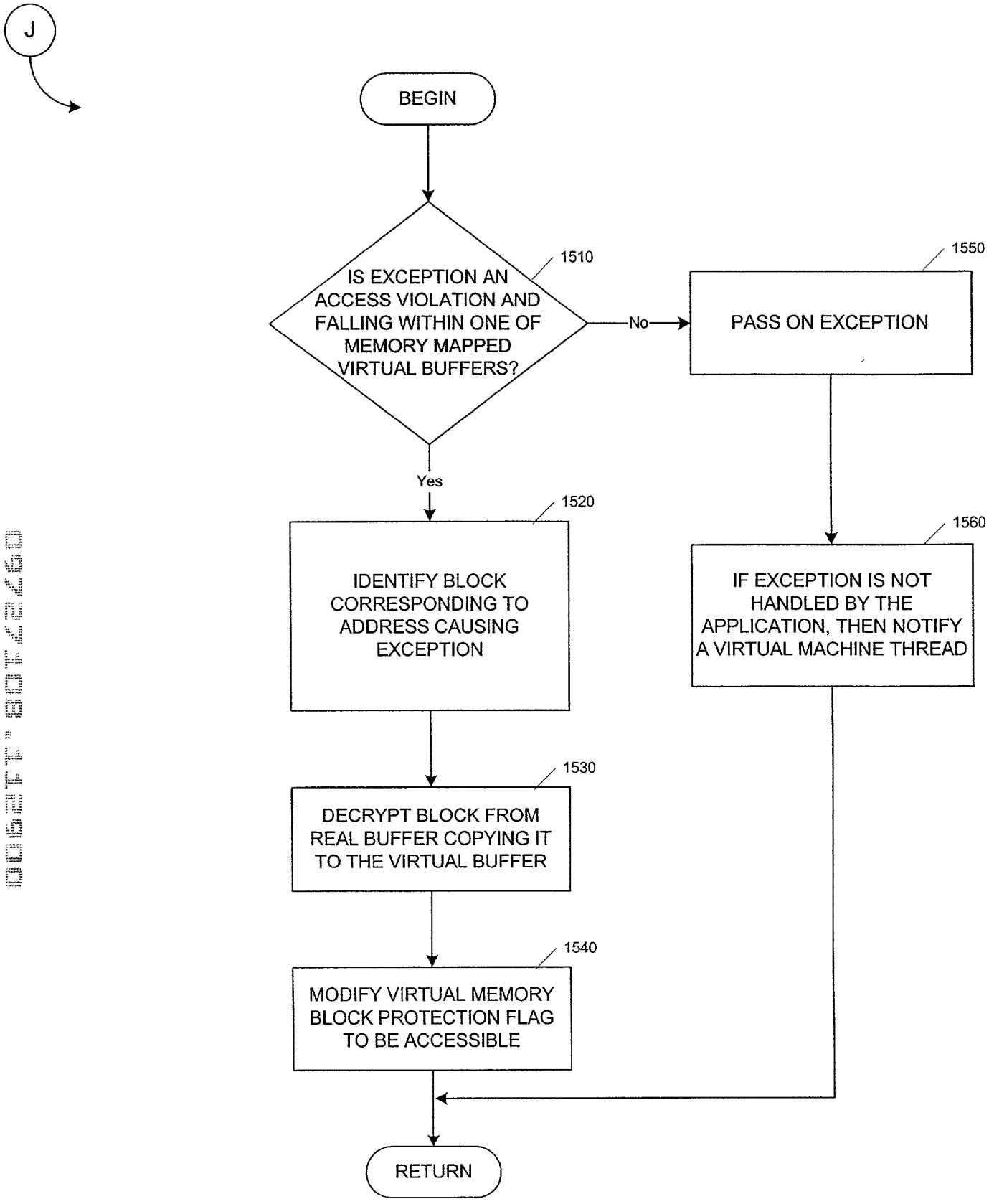
J
BEGIN
IS EXCEPTION AN ACCESS VIOLATION AND FALLING WITHIN ONE OF MEMORY MAPPED VIRTUAL BUFFERS?
1510
No
PASS ON EXCEPTION
1550
Yes
IDENTIFY BLOCK CORRESPONDING TO ADDRESS CAUSING EXCEPTION
1520
IF EXCEPTION IS NOT HANDLED BY THE APPLICATION, THEN NOTIFY A VIRTUAL MACHINE THREAD
1560
DECRYPT BLOCK FROM REAL BUFFER COPYING IT TO THE VIRTUAL BUFFER
1530
MODIFY VIRTUAL MEMORY BLOCK PROTECTION FLAG TO BE ACCESSIBLE
1540
RETURN

FIG. 15

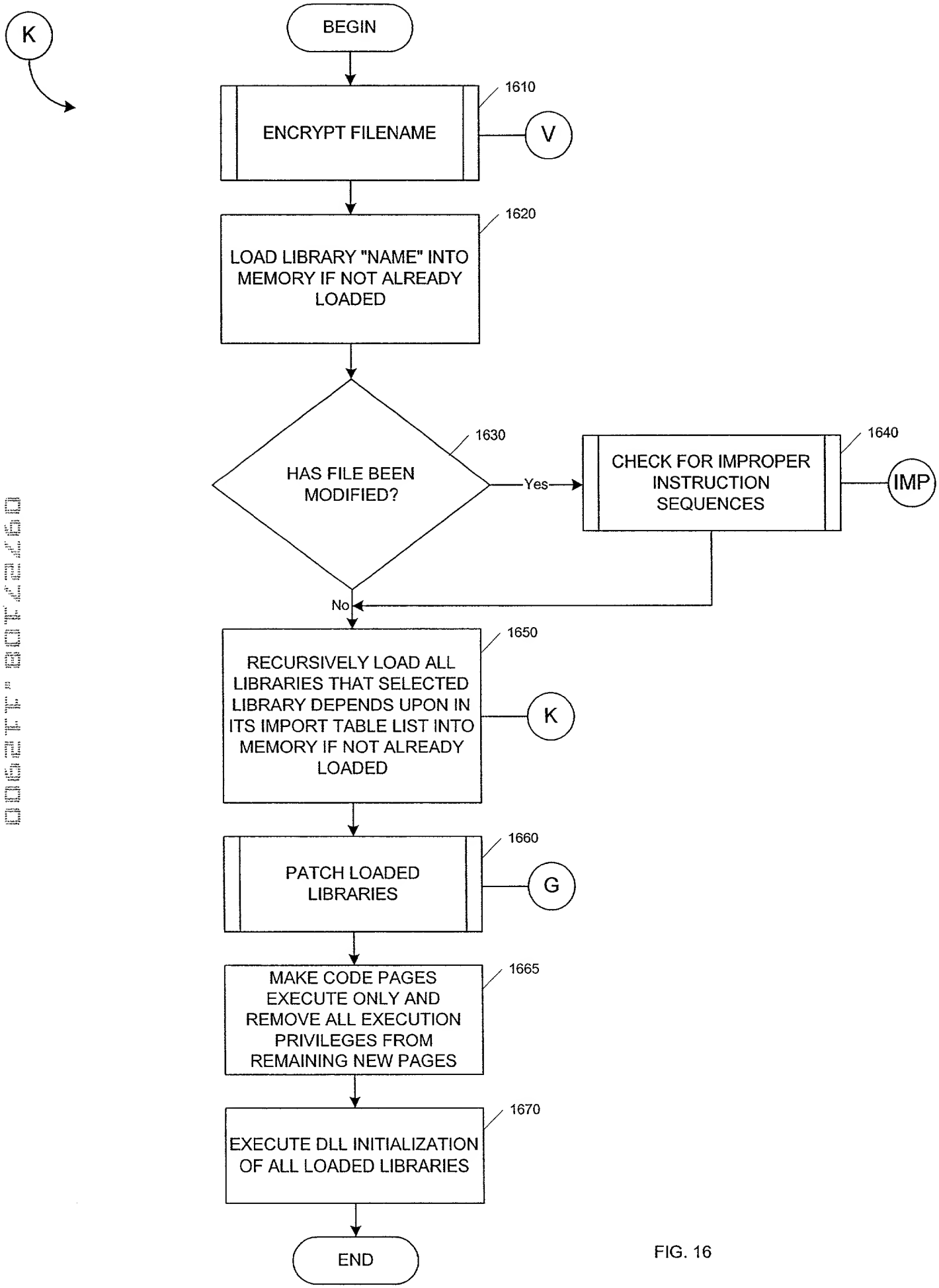
K
BEGIN
ENCRYPT FILENAME
1610
V
LOAD LIBRARY "NAME" INTO MEMORY IF NOT ALREADY LOADED
1620
HAS FILE BEEN MODIFIED?
1630
Yes
CHECK FOR IMPROPER INSTRUCTION SEQUENCES
1640
IMP
No
RECURSIVELY LOAD ALL LIBRARIES THAT SELECTED LIBRARY DEPENDS UPON IN ITS IMPORT TABLE LIST INTO MEMORY IF NOT ALREADY LOADED
1650
K
PATCH LOADED LIBRARIES
1660
G
MAKE CODE PAGES EXECUTE ONLY AND REMOVE ALL EXECUTION PRIVILEGES FROM REMAINING NEW PAGES
1665
EXECUTE DLL INITIALIZATION OF ALL LOADED LIBRARIES
1670
END

FIG. 16

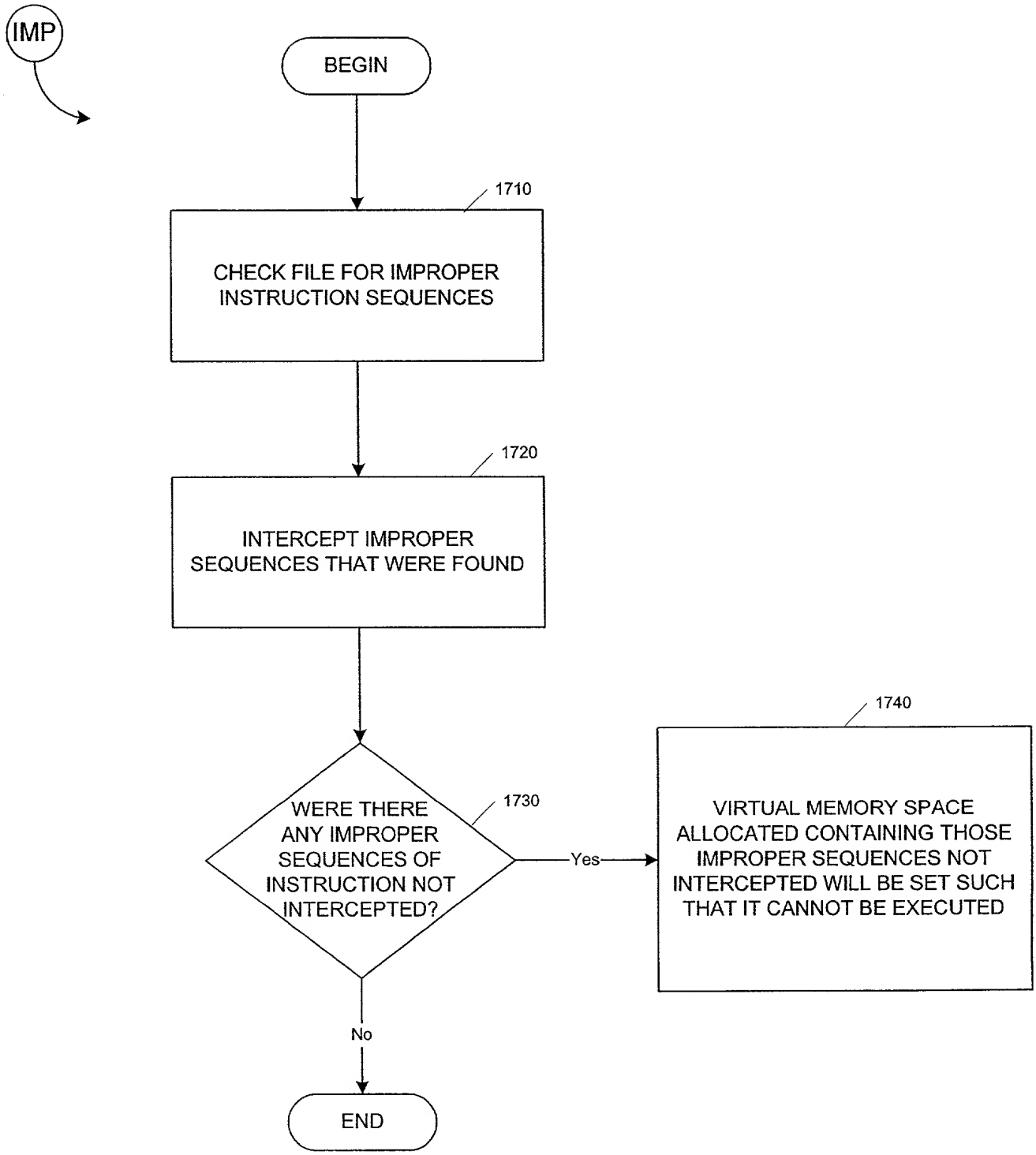
IMP
BEGIN
1710
CHECK FILE FOR IMPROPER INSTRUCTION SEQUENCES
1720
INTERCEPT IMPROPER SEQUENCES THAT WERE FOUND
1730
WERE THERE ANY IMPROPER SEQUENCES OF INSTRUCTION NOT INTERCEPTED?
Yes
1740
VIRTUAL MEMORY SPACE ALLOCATED CONTAINING THOSE IMPROPER SEQUENCES NOT INTERCEPTED WILL BE SET SUCH THAT IT CANNOT BE EXECUTED
No
END

FIG. 17

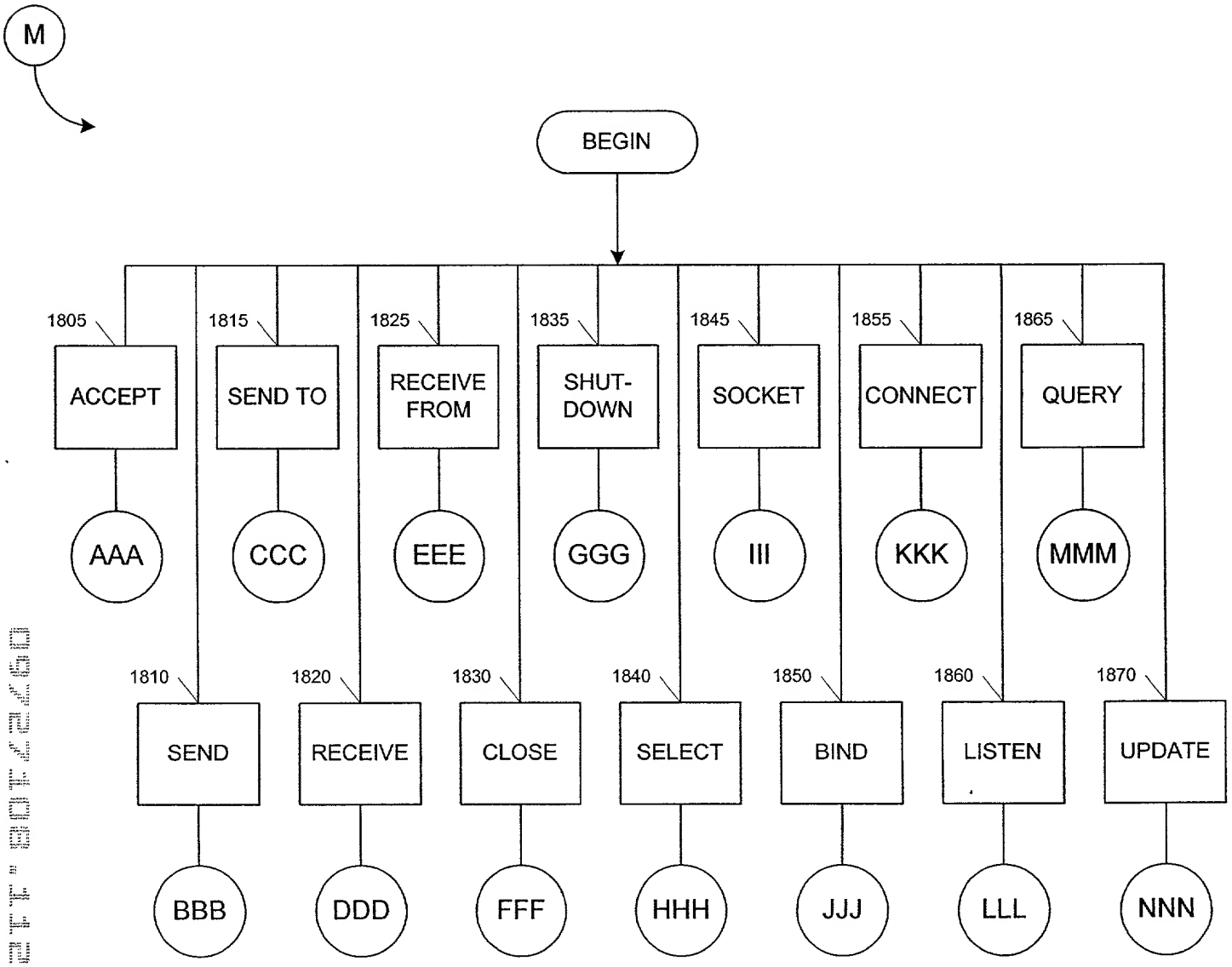
M
BEGIN
1805
ACCEPT
AAA
1810
SEND
BBB
1815
SEND TO
CCC
1820
RECEIVE
DDD
1825
RECEIVE FROM
EEE
1830
CLOSE
FFF
1835
SHUT-DOWN
GGG
1840
SELECT
HHH
1845
SOCKET
III
1850
BIND
JJJ
1855
CONNECT
KKK
1860
LISTEN
LLL
1865
QUERY
MMM
1870
UPDATE
NNN

FIG. 18

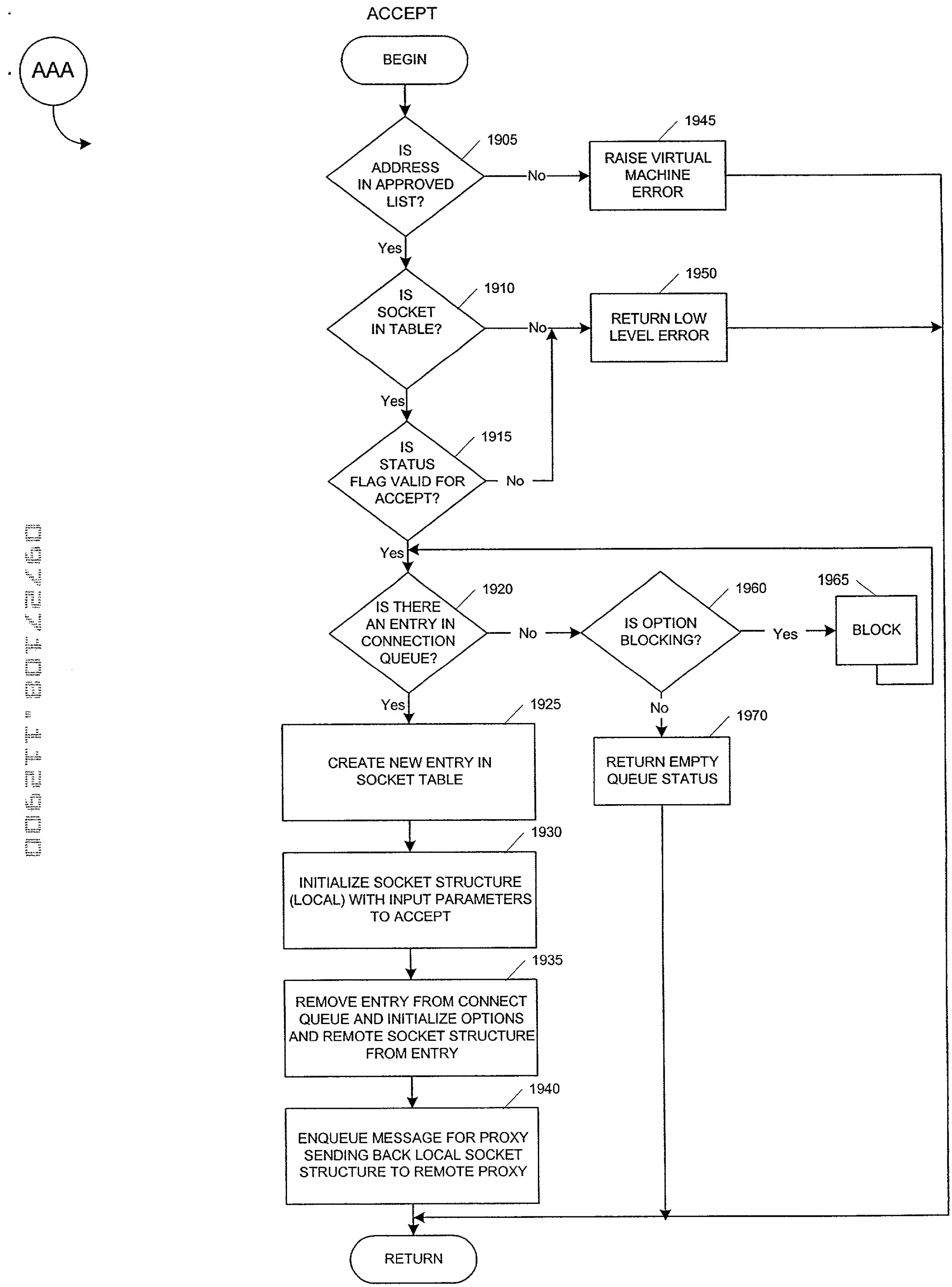

FIG. 19

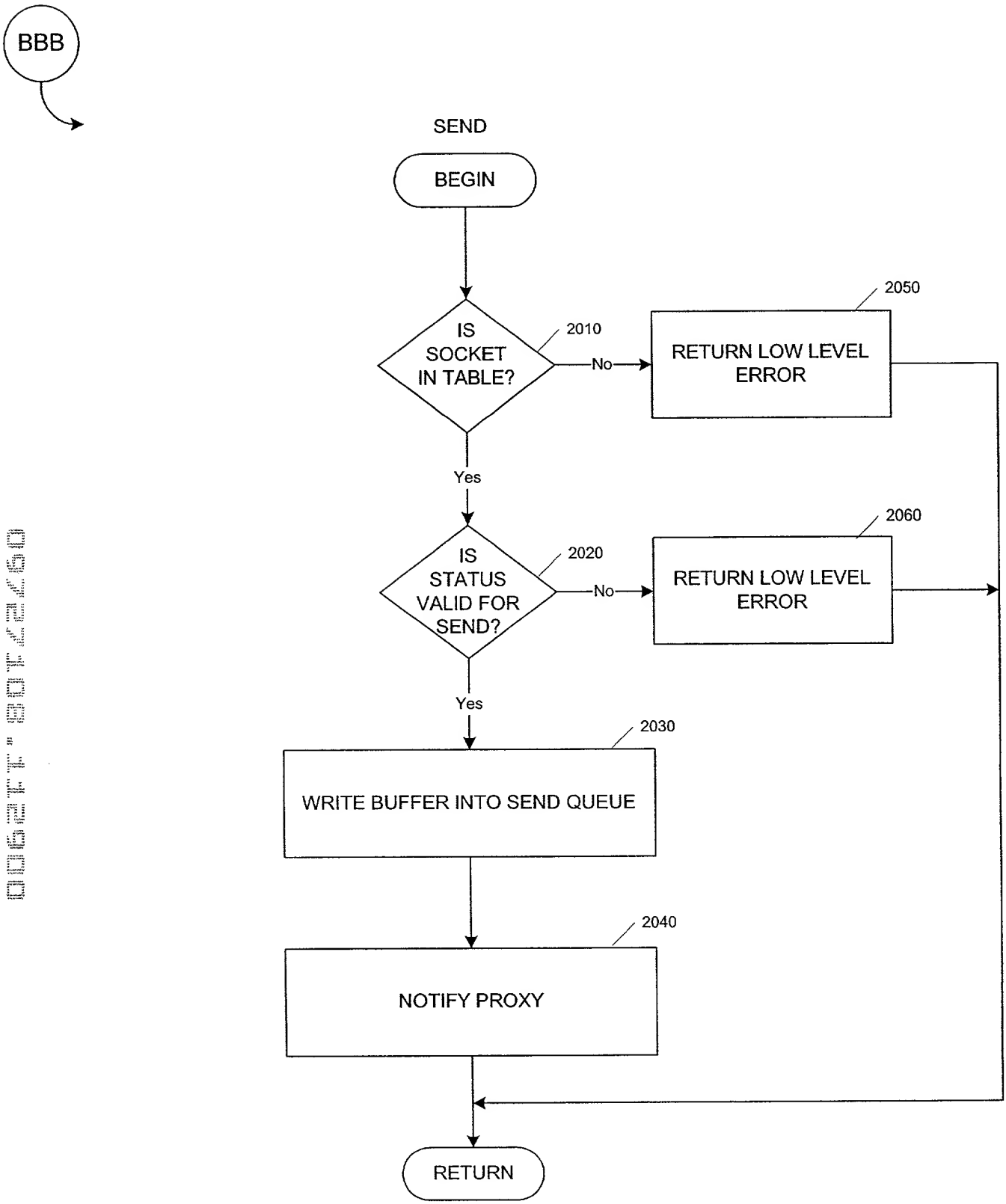
BBB
SEND
BEGIN
IS SOCKET IN TABLE?
2010
No
RETURN LOW LEVEL ERROR
2050
Yes
IS STATUS VALID FOR SEND?
2020
No
RETURN LOW LEVEL ERROR
2060
Yes
WRITE BUFFER INTO SEND QUEUE
2030
NOTIFY PROXY
2040
RETURN

FIG. 20

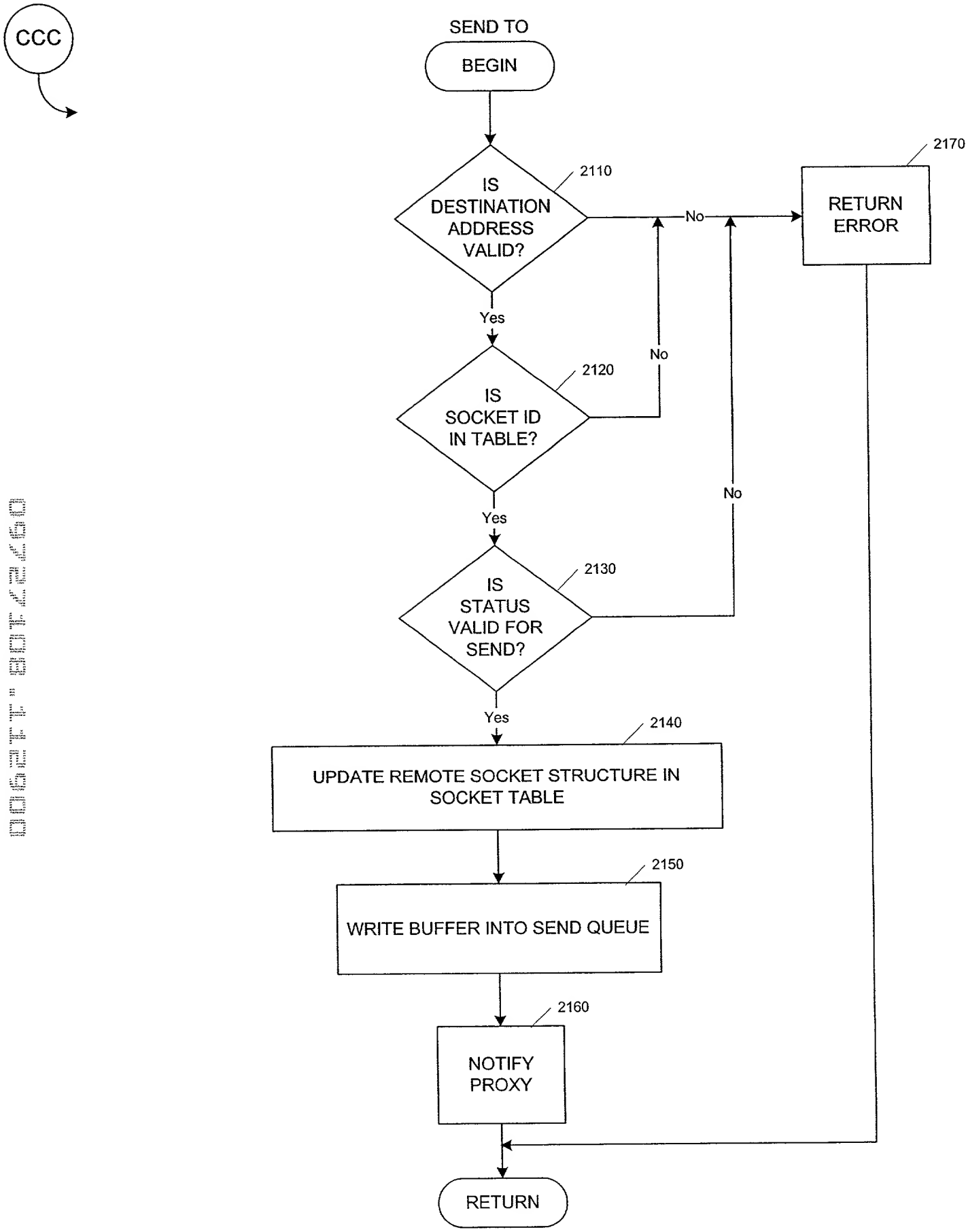
CCC
SEND TO
BEGIN
IS DESTINATION ADDRESS VALID?
2110
No
RETURN ERROR
2170
Yes
IS SOCKET ID IN TABLE?
2120
No
Yes
IS STATUS VALID FOR SEND?
2130
No
Yes
UPDATE REMOTE SOCKET STRUCTURE IN SOCKET TABLE
2140
WRITE BUFFER INTO SEND QUEUE
2150
NOTIFY PROXY
2160
RETURN

FIG. 21

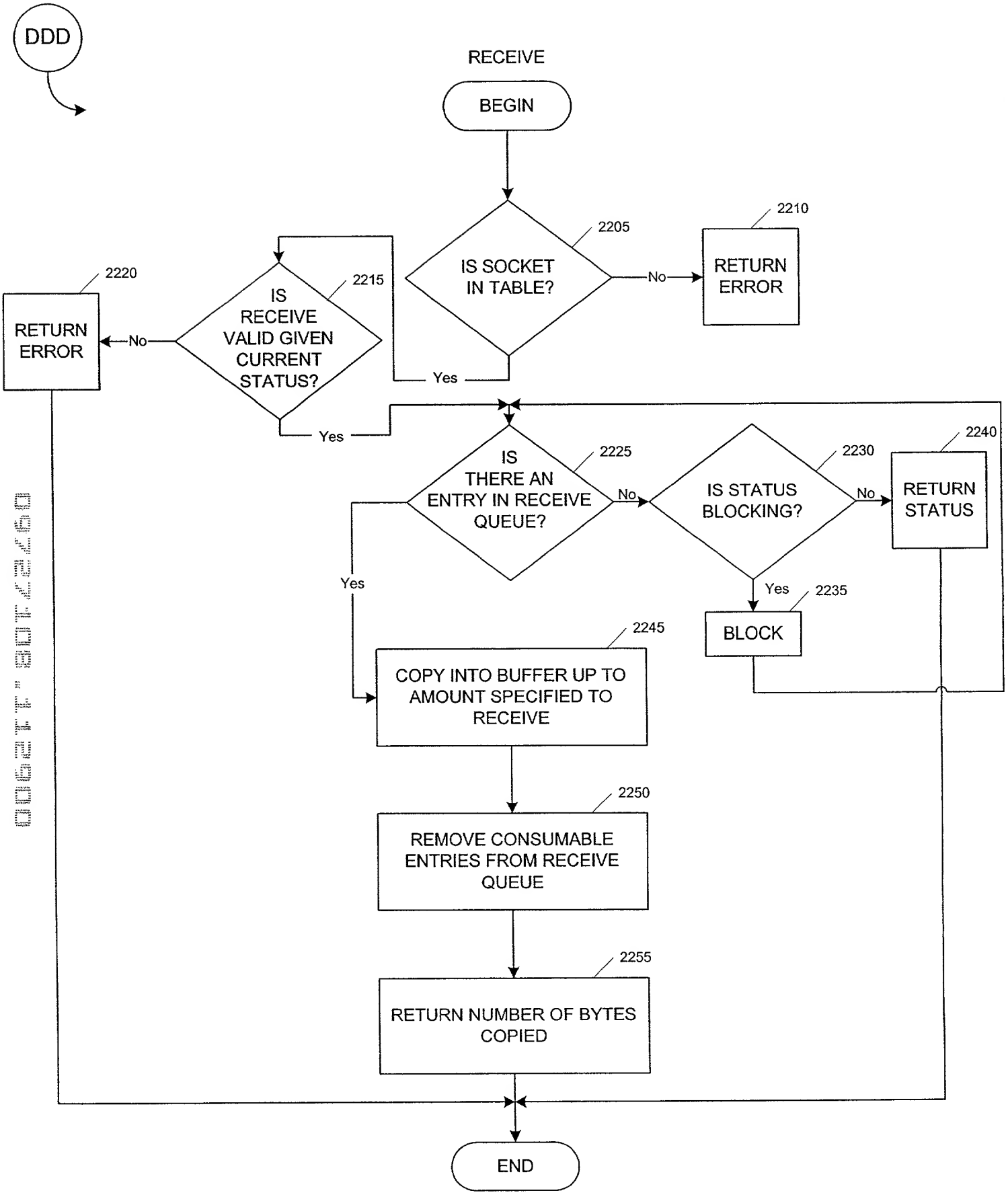
DDD
RECEIVE
BEGIN
2205
IS SOCKET IN TABLE?
No
2210
RETURN ERROR
Yes
2215
IS RECEIVE VALID GIVEN CURRENT STATUS?
No
2220
RETURN ERROR
Yes
2225
IS THERE AN ENTRY IN RECEIVE QUEUE?
No
2230
IS STATUS BLOCKING?
No
2240
RETURN STATUS
Yes
2235
BLOCK
Yes
2245
COPY INTO BUFFER UP TO AMOUNT SPECIFIED TO RECEIVE
2250
REMOVE CONSUMABLE ENTRIES FROM RECEIVE QUEUE
2255
RETURN NUMBER OF BYTES COPIED
END

FIG. 22

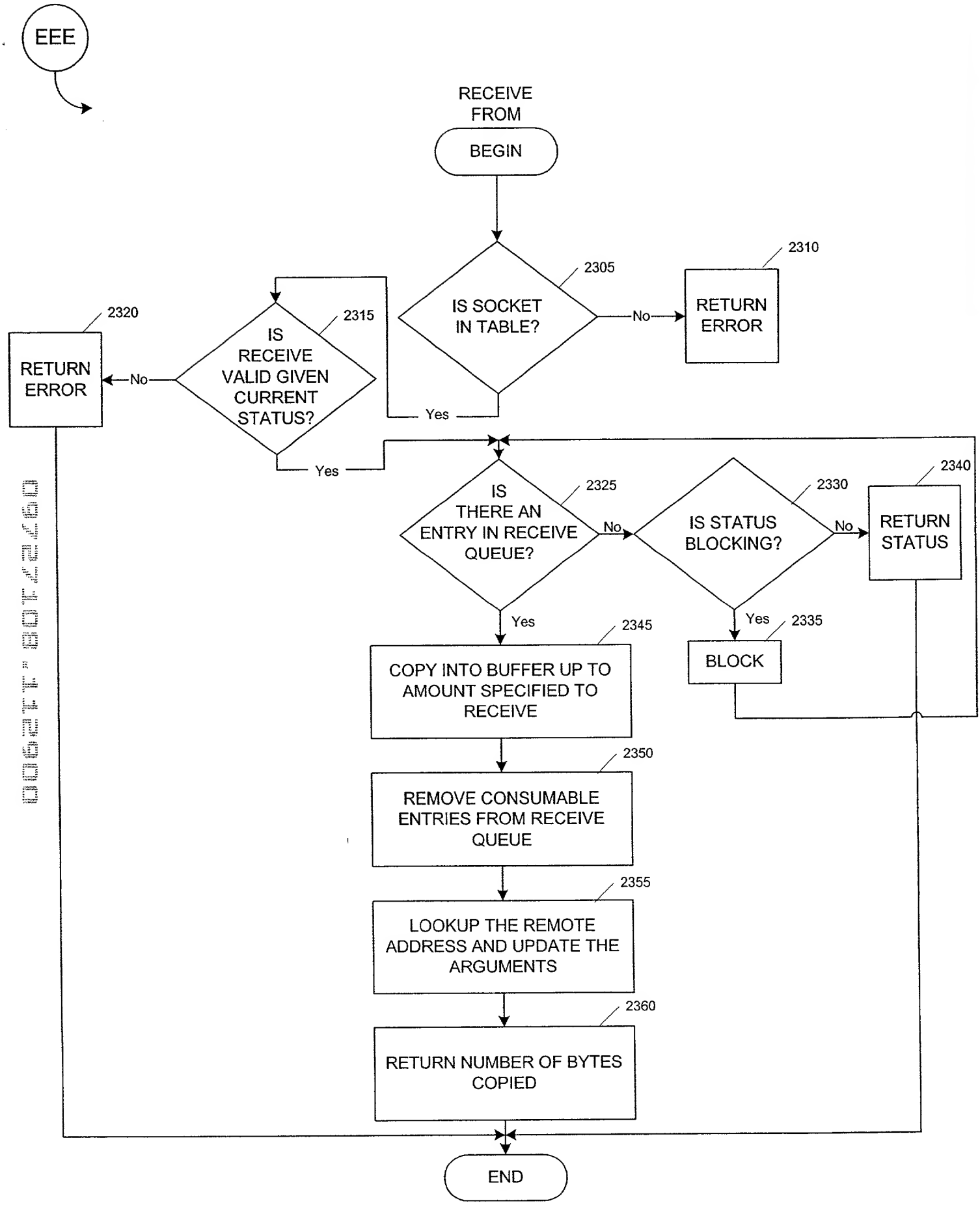
EEE
RECEIVE FROM
BEGIN
2305
IS SOCKET IN TABLE?
No
2310
RETURN ERROR
Yes
2315
IS RECEIVE VALID GIVEN CURRENT STATUS?
No
2320
RETURN ERROR
Yes
2325
IS THERE AN ENTRY IN RECEIVE QUEUE?
No
2330
IS STATUS BLOCKING?
No
2340
RETURN STATUS
Yes
2335
BLOCK
Yes
2345
COPY INTO BUFFER UP TO AMOUNT SPECIFIED TO RECEIVE
2350
REMOVE CONSUMABLE ENTRIES FROM RECEIVE QUEUE
2355
LOOKUP THE REMOTE ADDRESS AND UPDATE THE ARGUMENTS
2360
RETURN NUMBER OF BYTES COPIED
END

FIG. 23

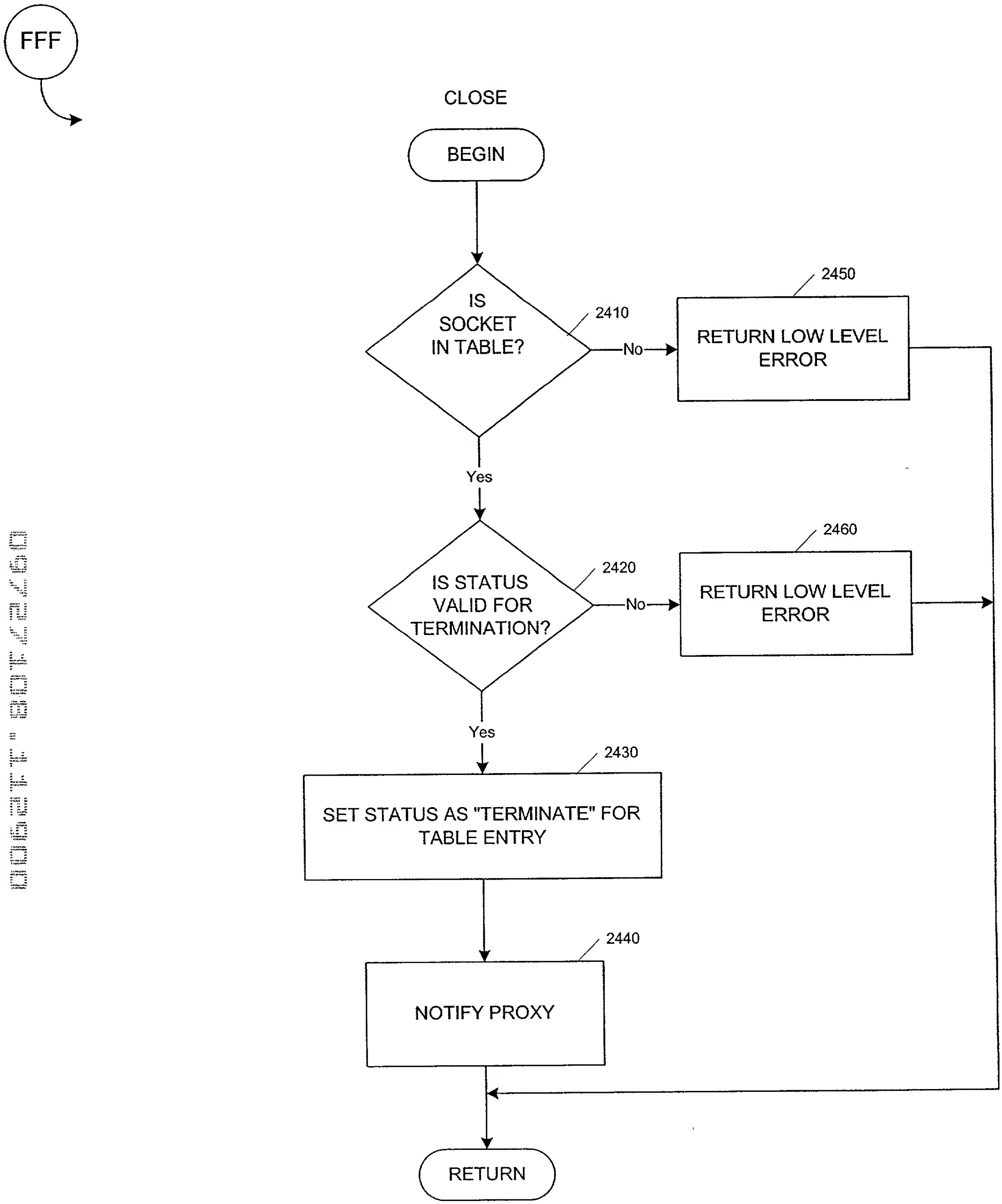
FFF
CLOSE
BEGIN
IS SOCKET IN TABLE?
2410
No
Yes
RETURN LOW LEVEL ERROR
2450
IS STATUS VALID FOR TERMINATION?
2420
No
Yes
RETURN LOW LEVEL ERROR
2460
SET STATUS AS "TERMINATE" FOR TABLE ENTRY
2430
NOTIFY PROXY
2440
RETURN

FIG. 24

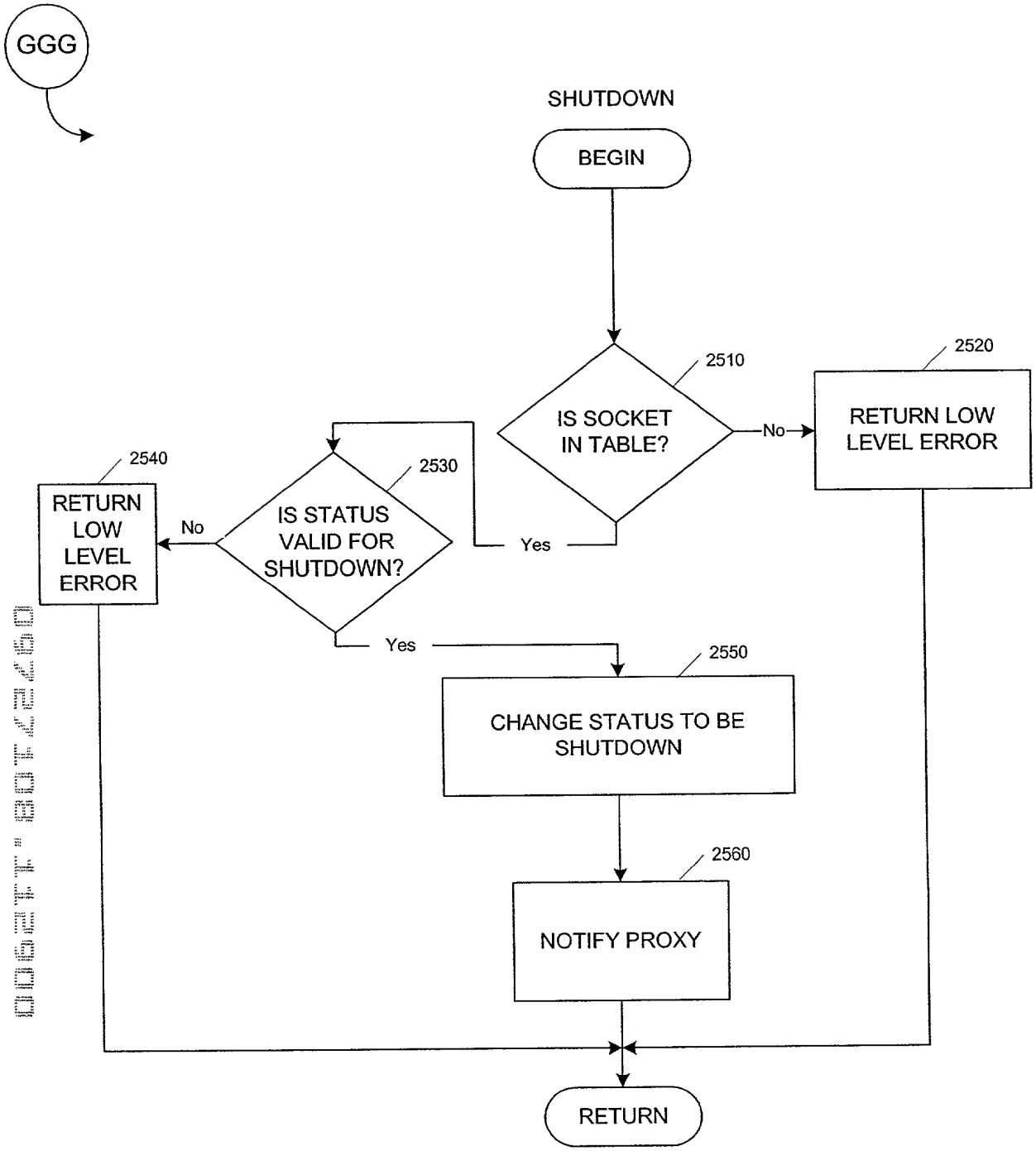
GGG
SHUTDOWN
BEGIN
2510
IS SOCKET IN TABLE?
No
2520
RETURN LOW LEVEL ERROR
Yes
2530
IS STATUS VALID FOR SHUTDOWN?
No
2540
RETURN LOW LEVEL ERROR
Yes
2550
CHANGE STATUS TO BE SHUTDOWN
2560
NOTIFY PROXY
RETURN

FIG. 25

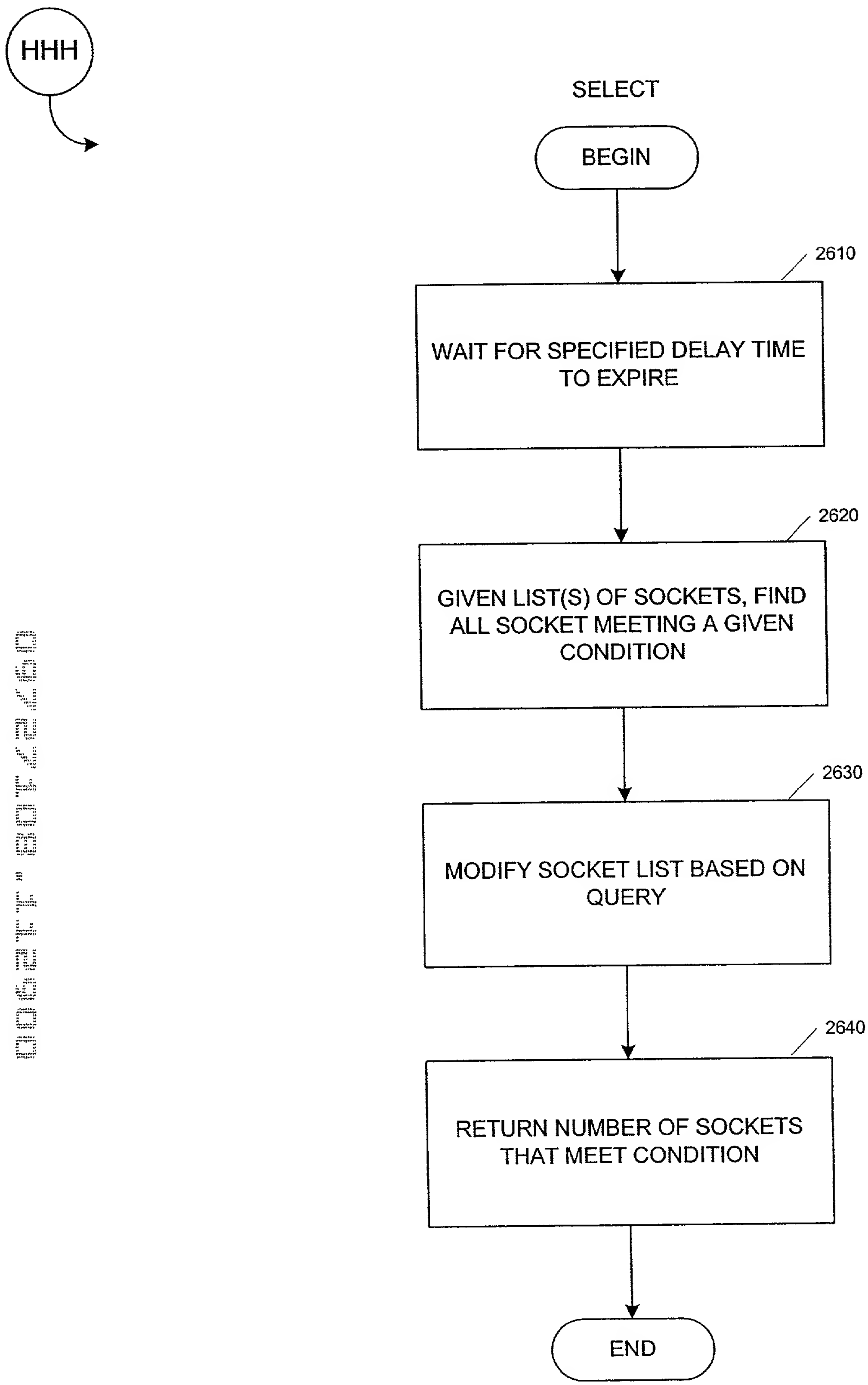
HHH
SELECT
BEGIN
2610
WAIT FOR SPECIFIED DELAY TIME TO EXPIRE
2620
GIVEN LIST(S) OF SOCKETS, FIND ALL SOCKET MEETING A GIVEN CONDITION
2630
MODIFY SOCKET LIST BASED ON QUERY
2640
RETURN NUMBER OF SOCKETS THAT MEET CONDITION
END

FIG. 26

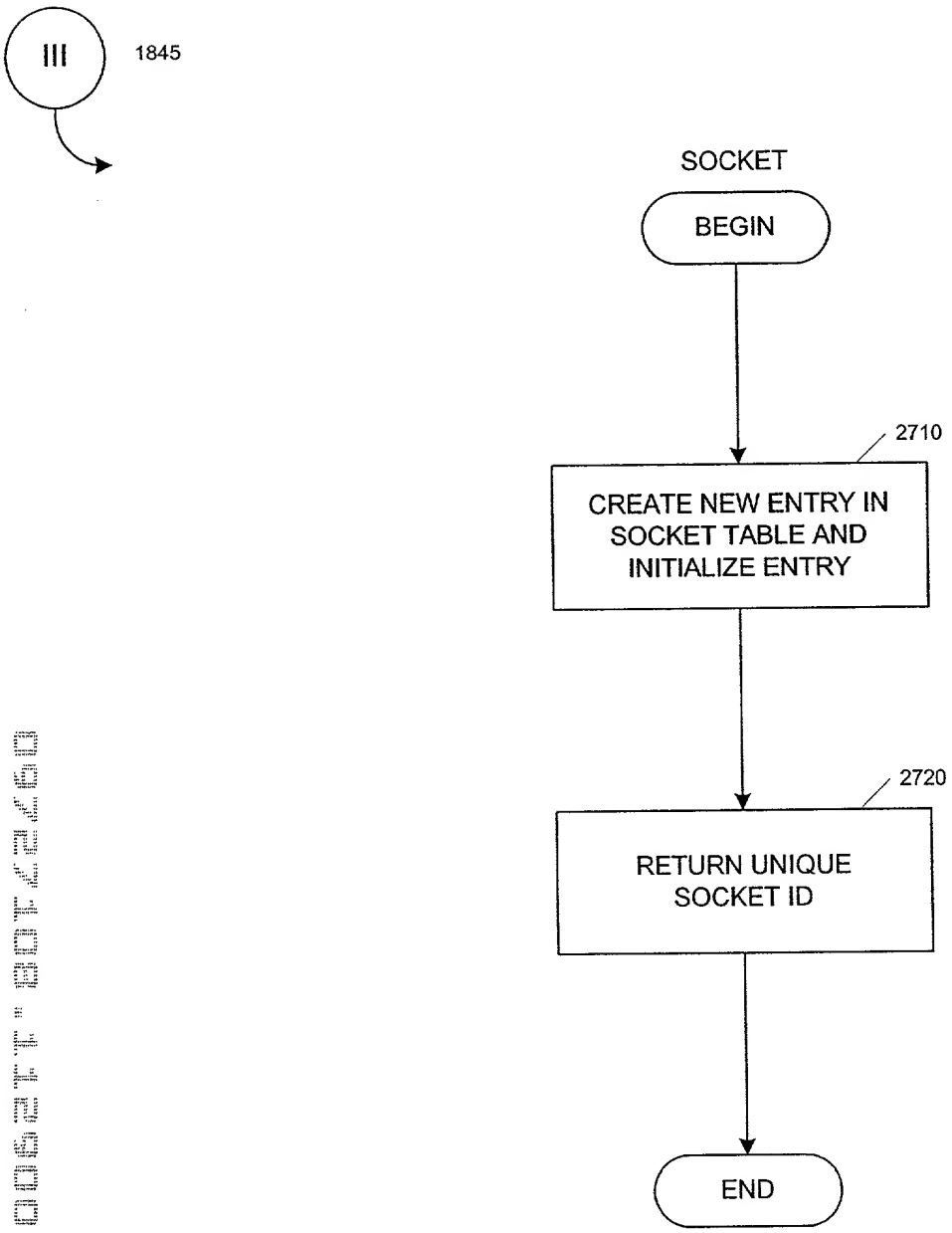
III
1845
SOCKET
BEGIN
2710
CREATE NEW ENTRY IN SOCKET TABLE AND INITIALIZE ENTRY
2720
RETURN UNIQUE SOCKET ID
END

FIG. 27

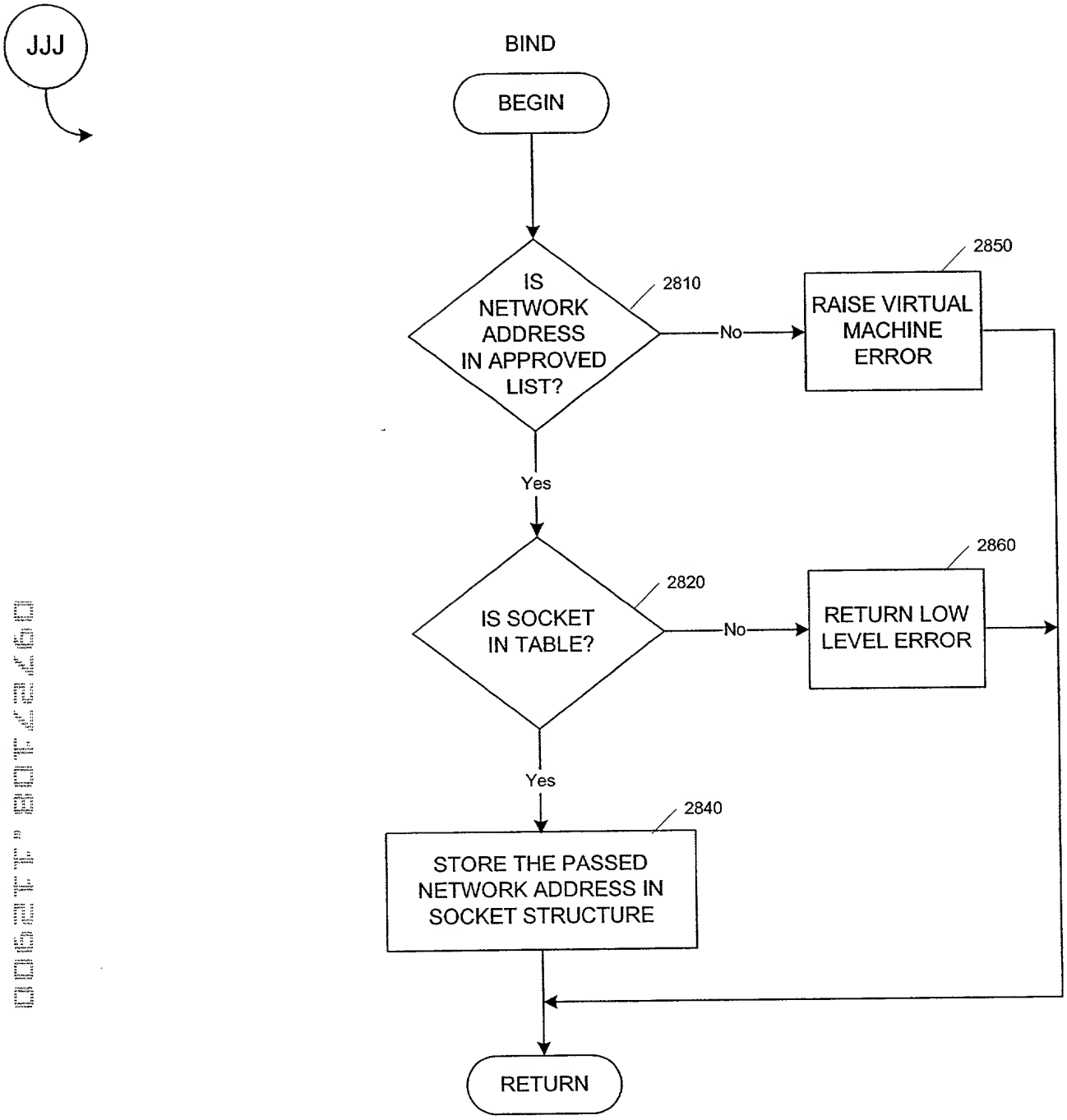
JJJ
BIND
BEGIN
IS NETWORK ADDRESS IN APPROVED LIST?
2810
No
RAISE VIRTUAL MACHINE ERROR
2850
Yes
IS SOCKET IN TABLE?
2820
No
RETURN LOW LEVEL ERROR
2860
Yes
STORE THE PASSED NETWORK ADDRESS IN SOCKET STRUCTURE
2840
RETURN

FIG. 28

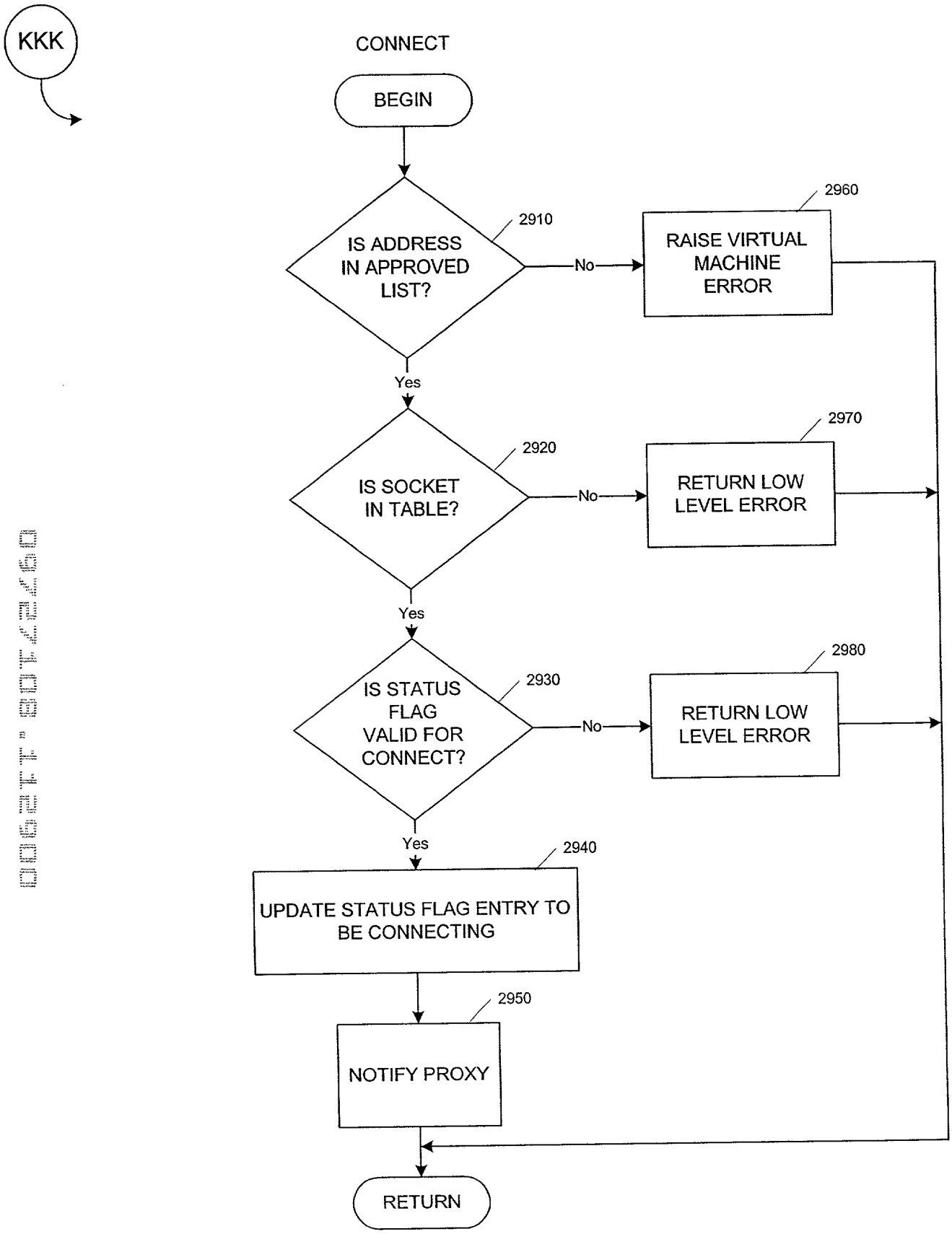
KKK
CONNECT
BEGIN
IS ADDRESS IN APPROVED LIST?
2910
No
RAISE VIRTUAL MACHINE ERROR
2960
Yes
IS SOCKET IN TABLE?
2920
No
RETURN LOW LEVEL ERROR
2970
Yes
IS STATUS FLAG VALID FOR CONNECT?
2930
No
RETURN LOW LEVEL ERROR
2980
Yes
UPDATE STATUS FLAG ENTRY TO BE CONNECTING
2940
NOTIFY PROXY
2950
RETURN

FIG. 29

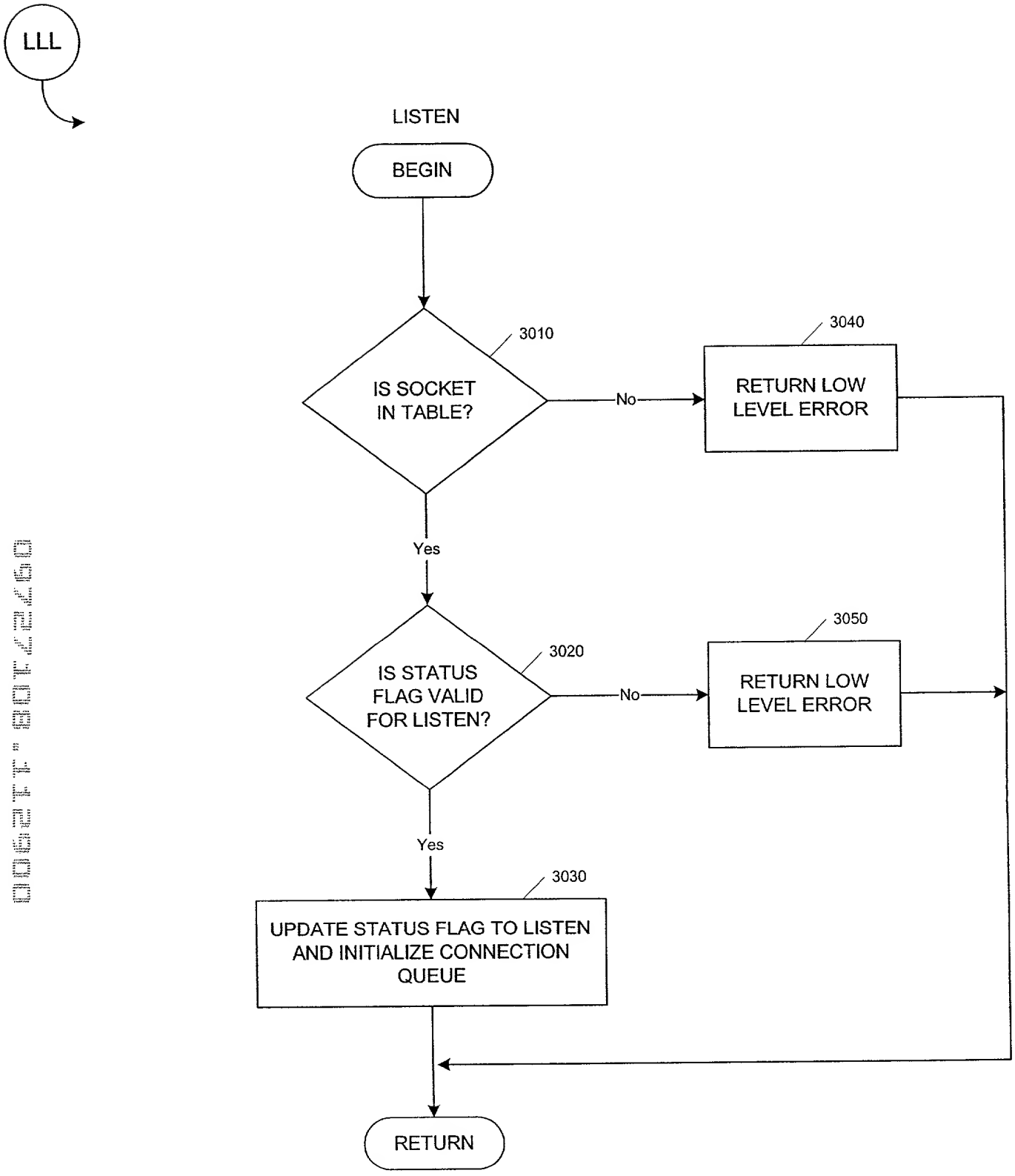
LLL
LISTEN
BEGIN
3010
IS SOCKET IN TABLE?
No
3040
RETURN LOW LEVEL ERROR
Yes
3020
IS STATUS FLAG VALID FOR LISTEN?
No
3050
RETURN LOW LEVEL ERROR
Yes
3030
UPDATE STATUS FLAG TO LISTEN AND INITIALIZE CONNECTION QUEUE
RETURN

FIG. 30

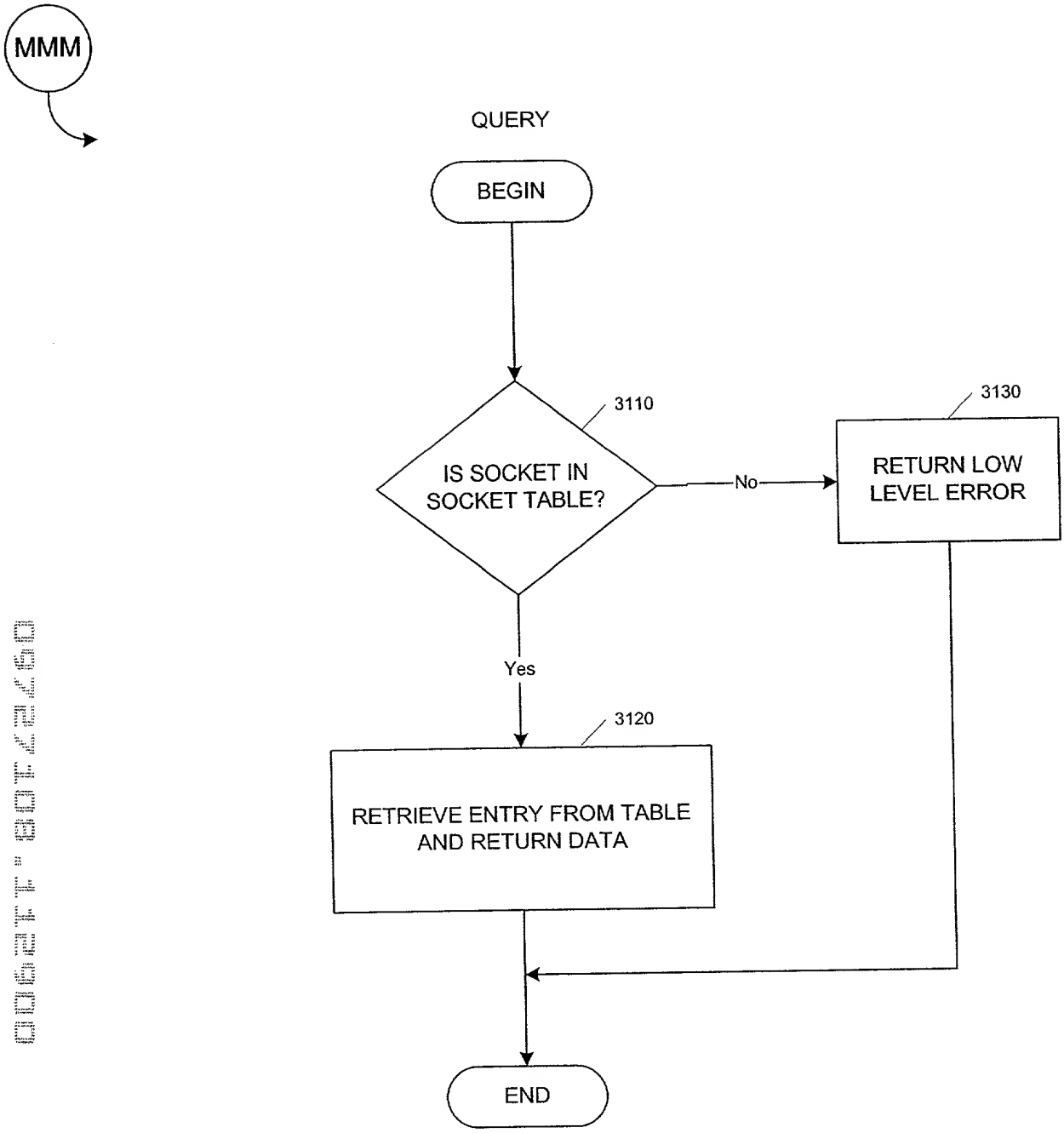
MMM
QUERY
BEGIN
3110
IS SOCKET IN SOCKET TABLE?
No
3130
RETURN LOW LEVEL ERROR
Yes
3120
RETRIEVE ENTRY FROM TABLE AND RETURN DATA
END

FIG. 31

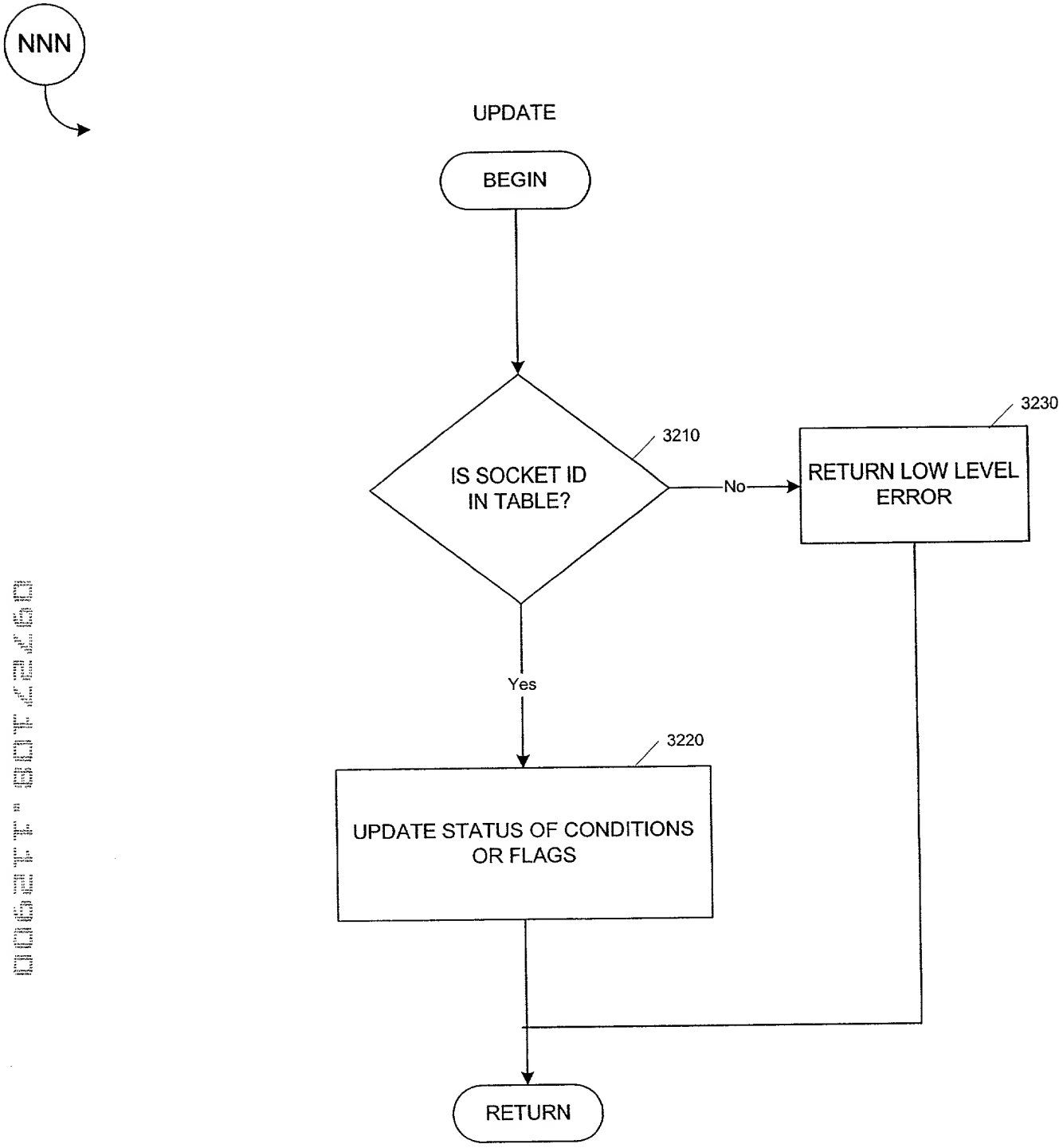
NNN
UPDATE
BEGIN
IS SOCKET ID IN TABLE?
3210
No
RETURN LOW LEVEL ERROR
3230
Yes
UPDATE STATUS OF CONDITIONS OR FLAGS
3220
RETURN

FIG. 32

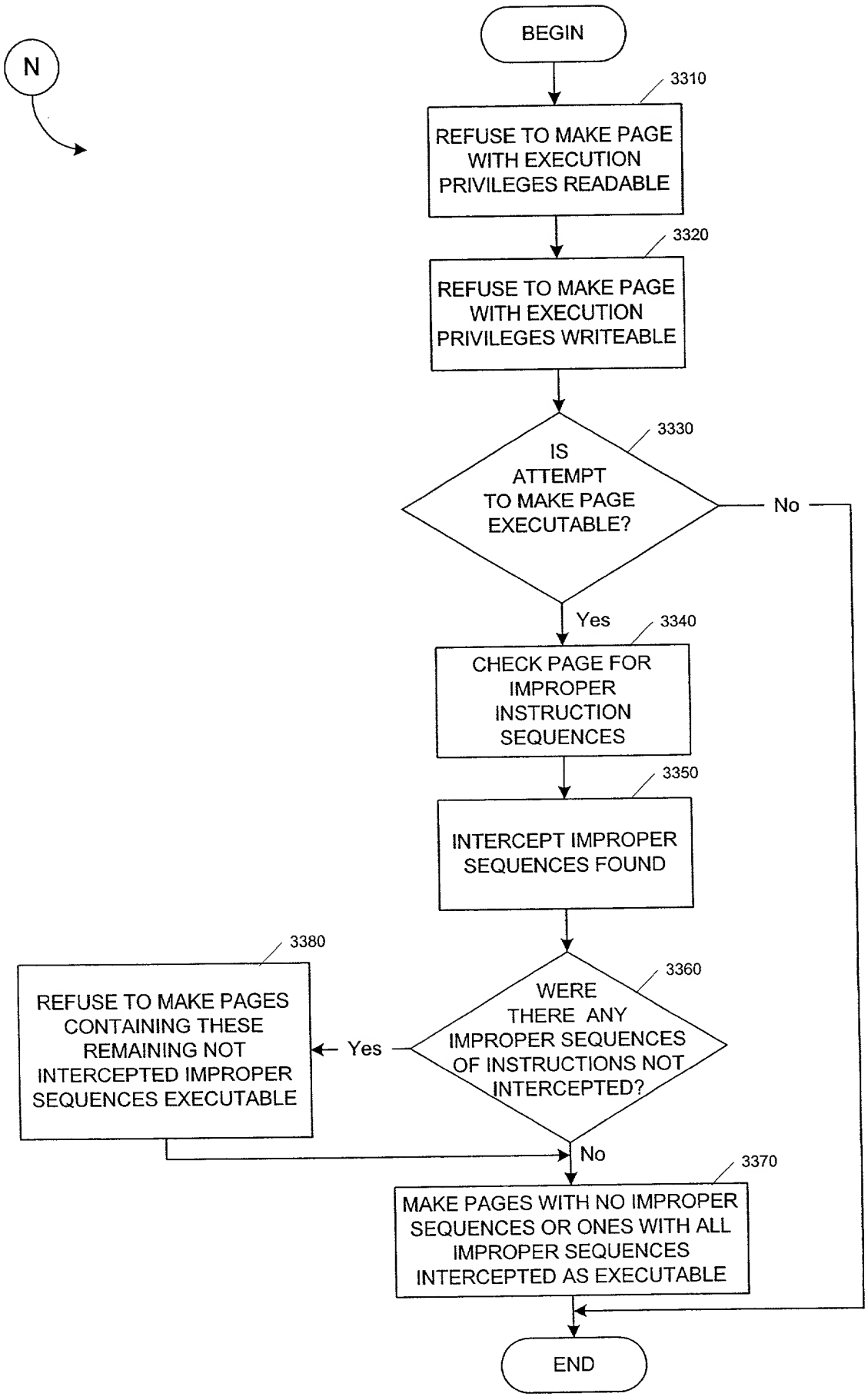

FIG. 33

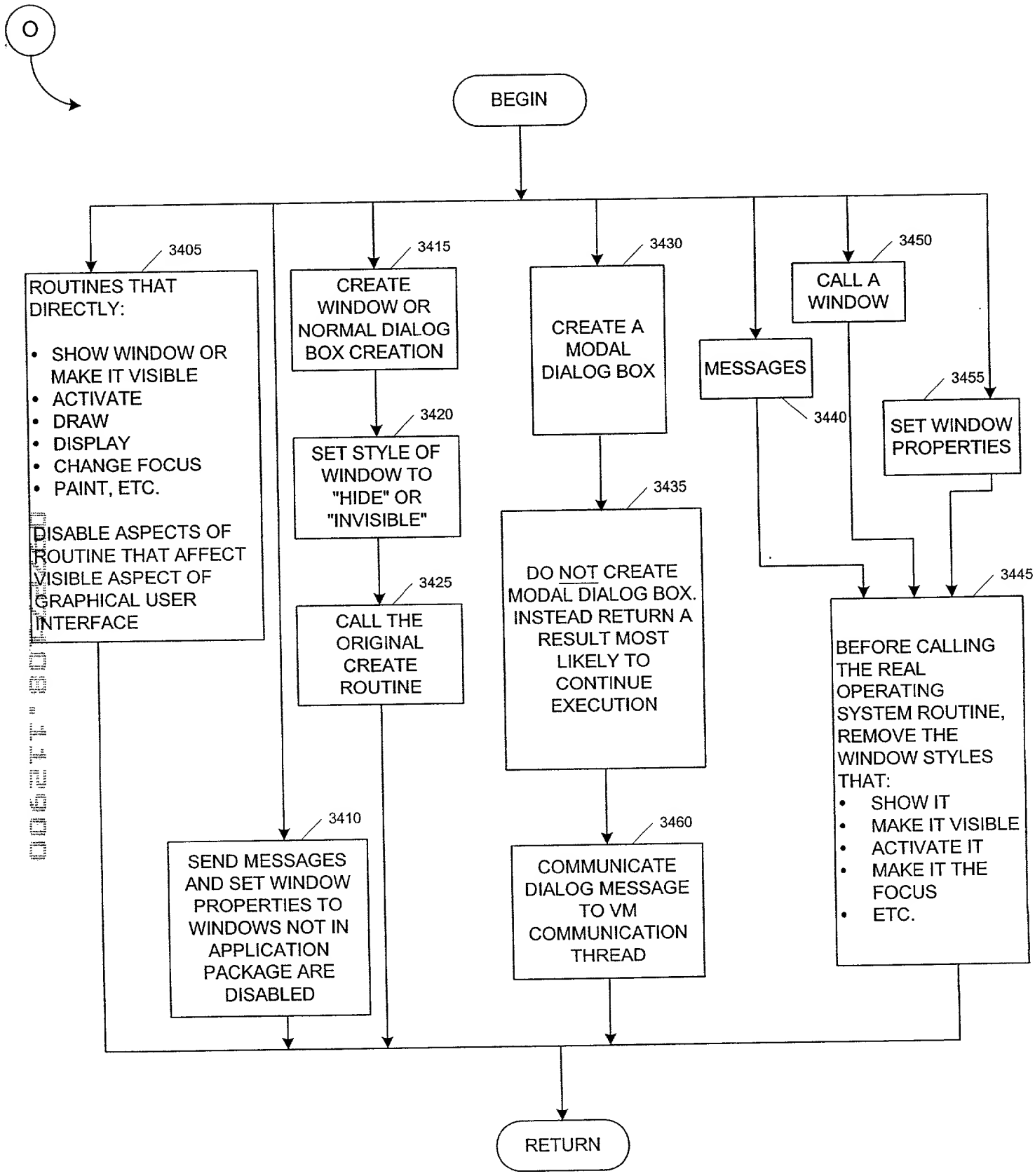
O
BEGIN
3405
ROUTINES THAT DIRECTLY:
• SHOW WINDOW OR MAKE IT VISIBLE
• ACTIVATE
• DRAW
• DISPLAY
• CHANGE FOCUS
• PAINT, ETC.
DISABLE ASPECTS OF ROUTINE THAT AFFECT VISIBLE ASPECT OF GRAPHICAL USER INTERFACE
3415
CREATE WINDOW OR NORMAL DIALOG BOX CREATION
3420
SET STYLE OF WINDOW TO "HIDE" OR "INVISIBLE"
3425
CALL THE ORIGINAL CREATE ROUTINE
3410
SEND MESSAGES AND SET WINDOW PROPERTIES TO WINDOWS NOT IN APPLICATION PACKAGE ARE DISABLED
3430
CREATE A MODAL DIALOG BOX
3435
DO NOT CREATE MODAL DIALOG BOX. INSTEAD RETURN A RESULT MOST LIKELY TO CONTINUE EXECUTION
3460
COMMUNICATE DIALOG MESSAGE TO VM COMMUNICATION THREAD
MESSAGES
3440
3450
CALL A WINDOW
3455
SET WINDOW PROPERTIES
3445
BEFORE CALLING THE REAL OPERATING SYSTEM ROUTINE, REMOVE THE WINDOW STYLES THAT:
• SHOW IT
• MAKE IT VISIBLE
• ACTIVATE IT
• MAKE IT THE FOCUS
• ETC.
RETURN

FIG. 34

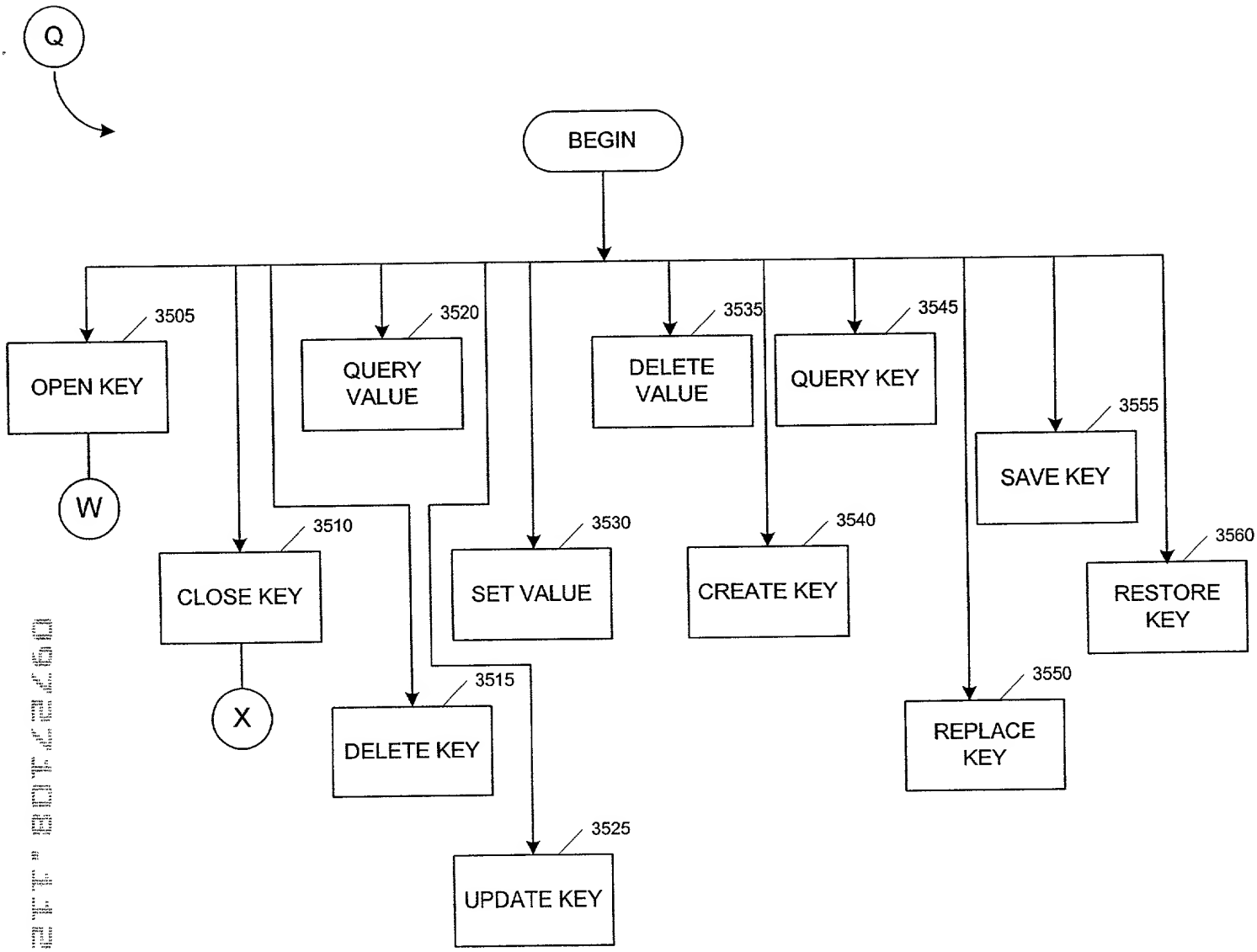
Q
BEGIN
3505
OPEN KEY
W
3510
CLOSE KEY
X
3515
DELETE KEY
3520
QUERY VALUE
3525
UPDATE KEY
3530
SET VALUE
3535
DELETE VALUE
3540
CREATE KEY
3545
QUERY KEY
3550
REPLACE KEY
3555
SAVE KEY
3560
RESTORE KEY

FIG. 35

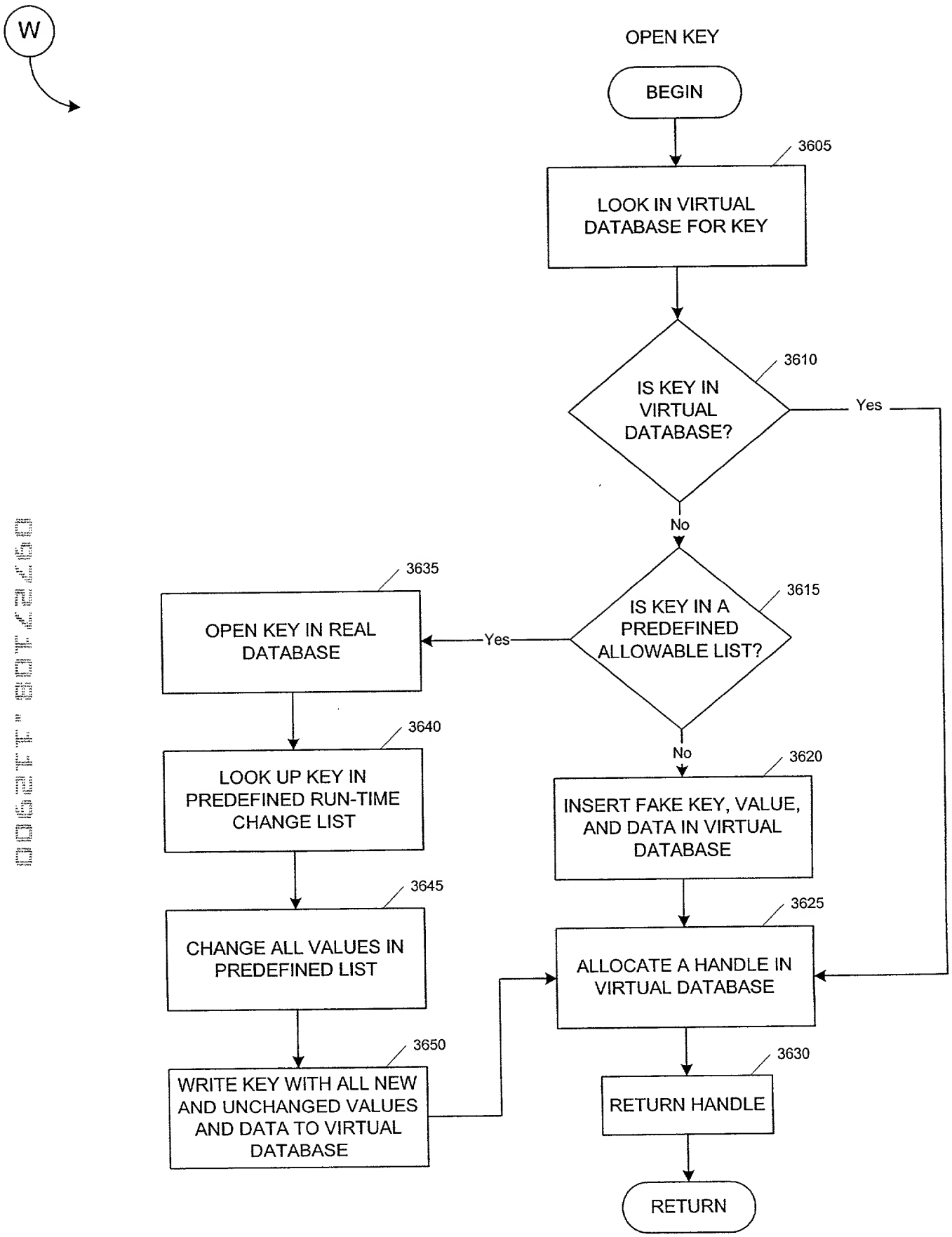
W
OPEN KEY
BEGIN
3605
LOOK IN VIRTUAL DATABASE FOR KEY
3610
IS KEY IN VIRTUAL DATABASE?
Yes
No
3615
IS KEY IN A PREDEFINED ALLOWABLE LIST?
Yes
3635
OPEN KEY IN REAL DATABASE
No
3620
INSERT FAKE KEY, VALUE, AND DATA IN VIRTUAL DATABASE
3640
LOOK UP KEY IN PREDEFINED RUN-TIME CHANGE LIST
3645
CHANGE ALL VALUES IN PREDEFINED LIST
3650
WRITE KEY WITH ALL NEW AND UNCHANGED VALUES AND DATA TO VIRTUAL DATABASE
3625
ALLOCATE A HANDLE IN VIRTUAL DATABASE
3630
RETURN HANDLE
RETURN

FIG. 36

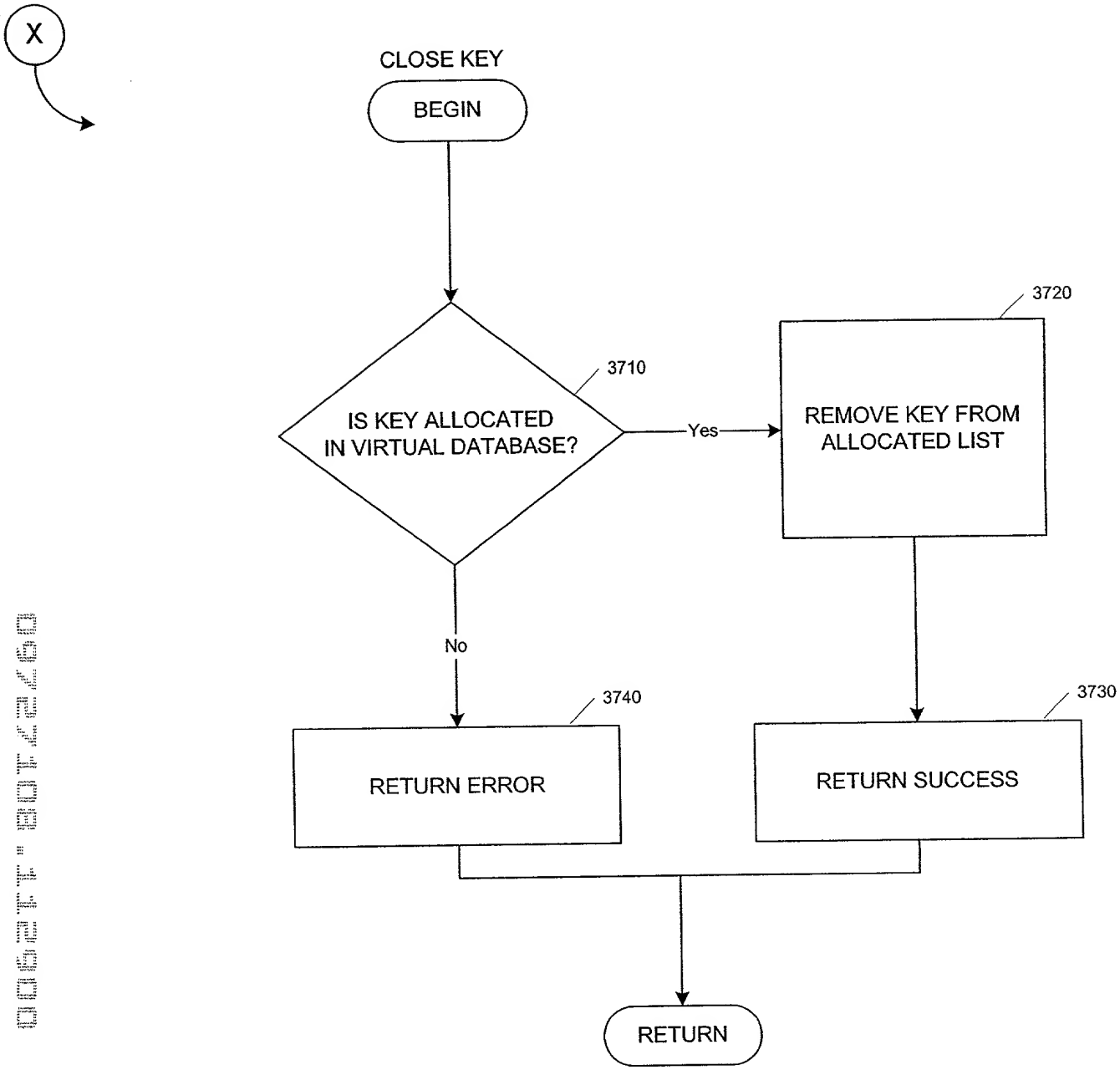
X
CLOSE KEY
BEGIN
IS KEY ALLOCATED IN VIRTUAL DATABASE?
3710
Yes
REMOVE KEY FROM ALLOCATED LIST
3720
No
RETURN ERROR
3740
RETURN SUCCESS
3730
RETURN

FIG. 37

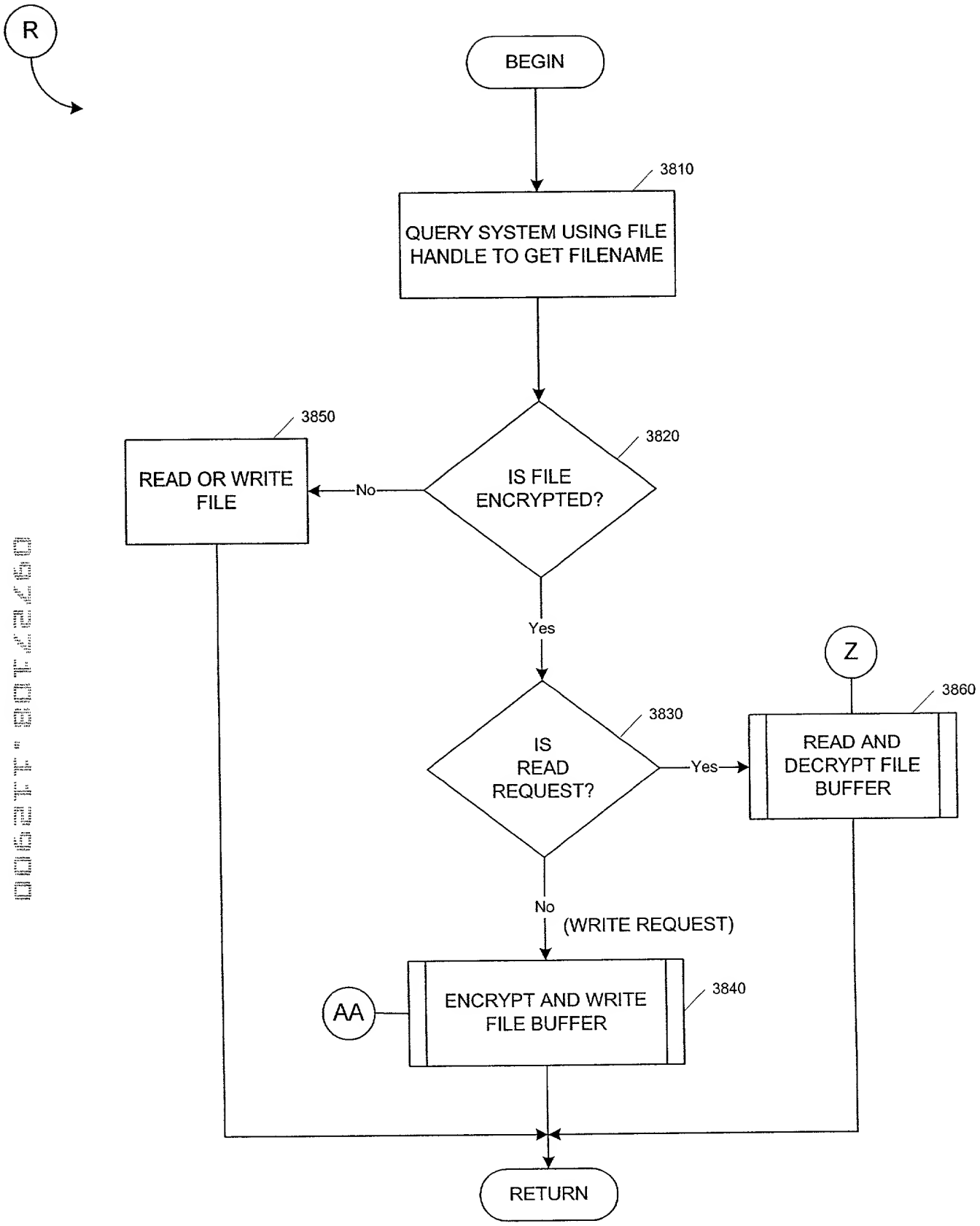
R
BEGIN
3810
QUERY SYSTEM USING FILE HANDLE TO GET FILENAME
3820
IS FILE ENCRYPTED?
No
3850
READ OR WRITE FILE
Yes
Z
3860
3830
IS READ REQUEST?
Yes
READ AND DECRYPT FILE BUFFER
No
(WRITE REQUEST)
AA
ENCRYPT AND WRITE FILE BUFFER
3840
RETURN

FIG. 38

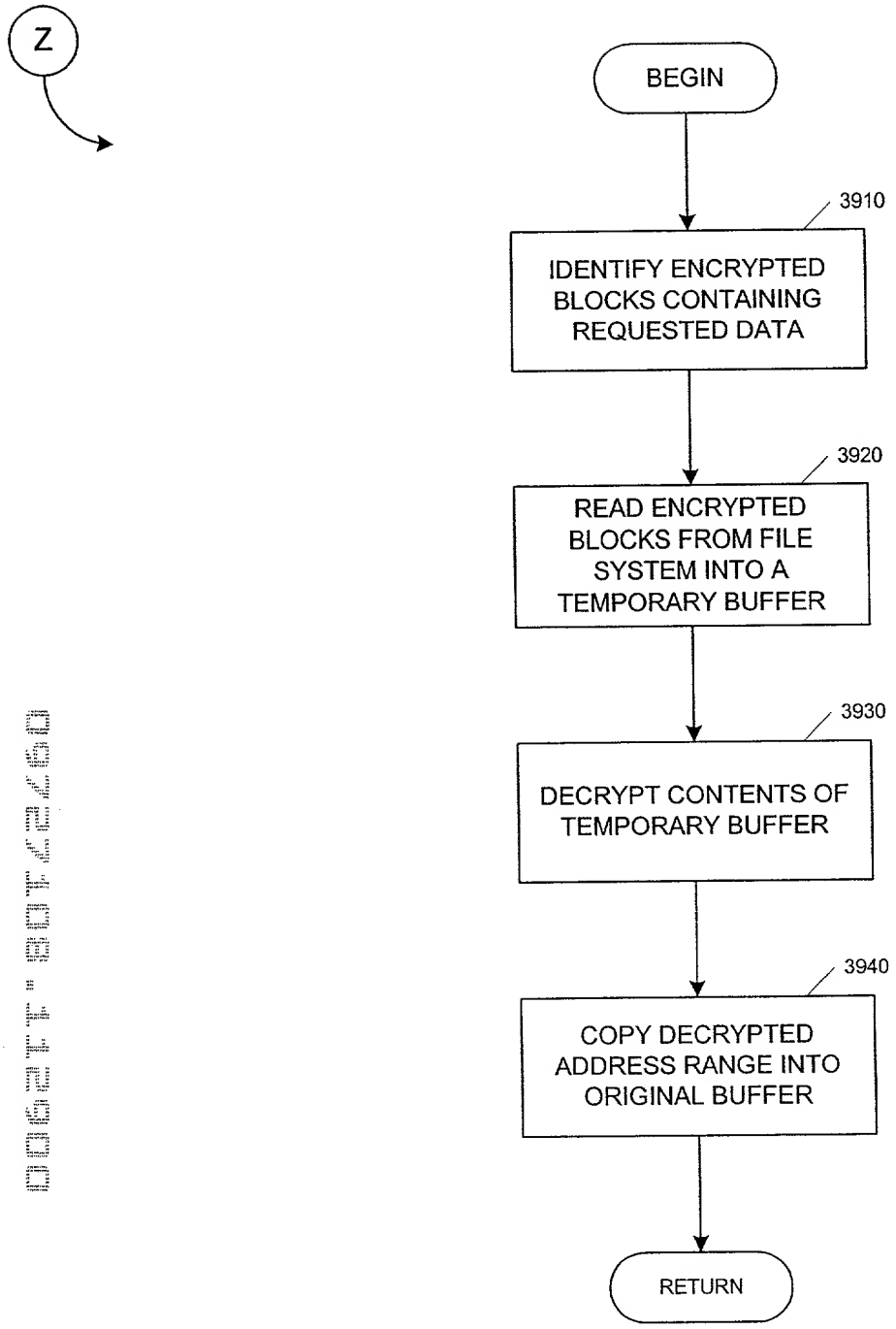
Z
BEGIN
3910
IDENTIFY ENCRYPTED BLOCKS CONTAINING REQUESTED DATA
3920
READ ENCRYPTED BLOCKS FROM FILE SYSTEM INTO A TEMPORARY BUFFER
3930
DECRYPT CONTENTS OF TEMPORARY BUFFER
3940
COPY DECRYPTED ADDRESS RANGE INTO ORIGINAL BUFFER
RETURN

FIG. 39

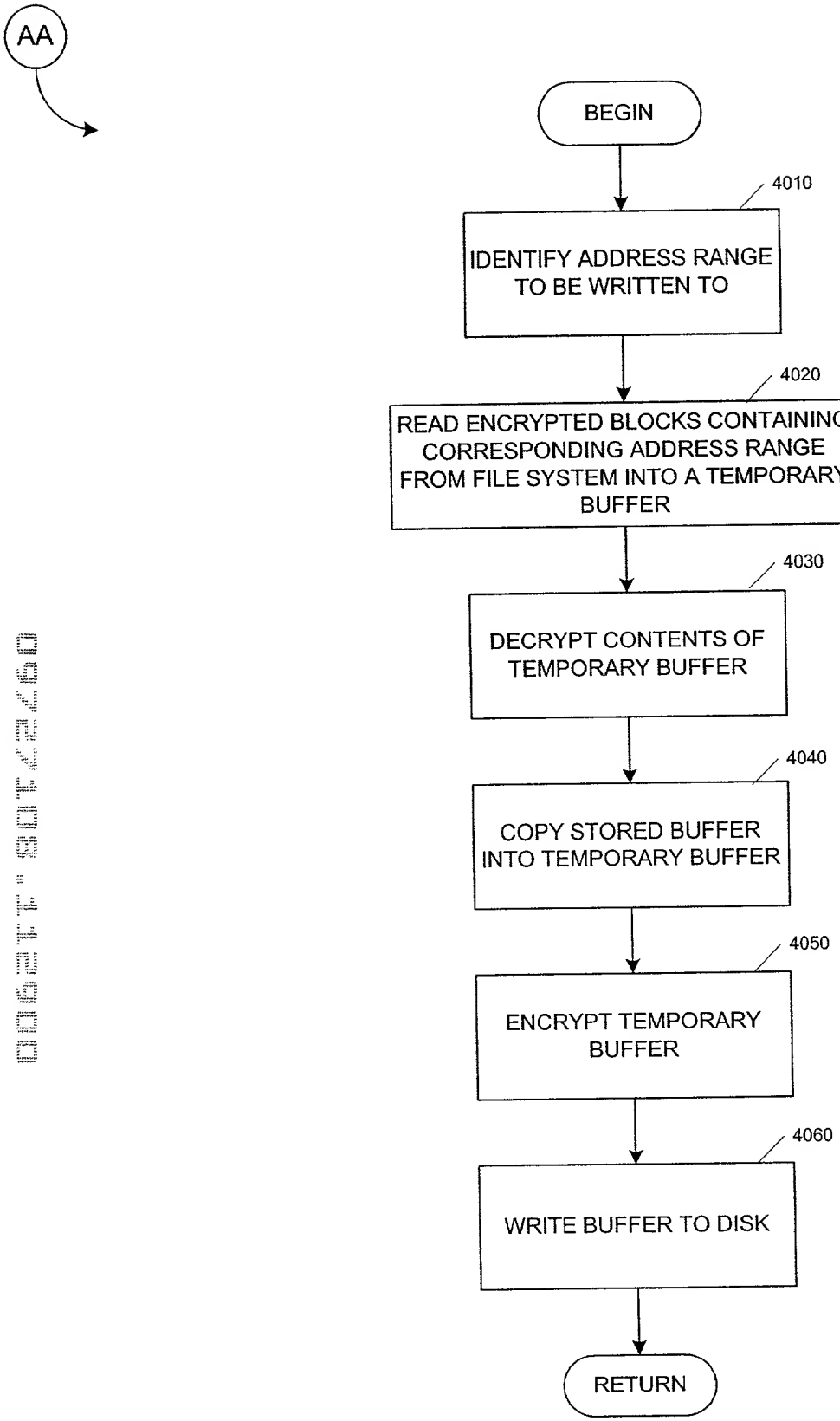
AA
BEGIN
4010
IDENTIFY ADDRESS RANGE TO BE WRITTEN TO
4020
READ ENCRYPTED BLOCKS CONTAINING CORRESPONDING ADDRESS RANGE FROM FILE SYSTEM INTO A TEMPORARY BUFFER
4030
DECRYPT CONTENTS OF TEMPORARY BUFFER
4040
COPY STORED BUFFER INTO TEMPORARY BUFFER
4050
ENCRYPT TEMPORARY BUFFER
4060
WRITE BUFFER TO DISK
RETURN

FIG. 40

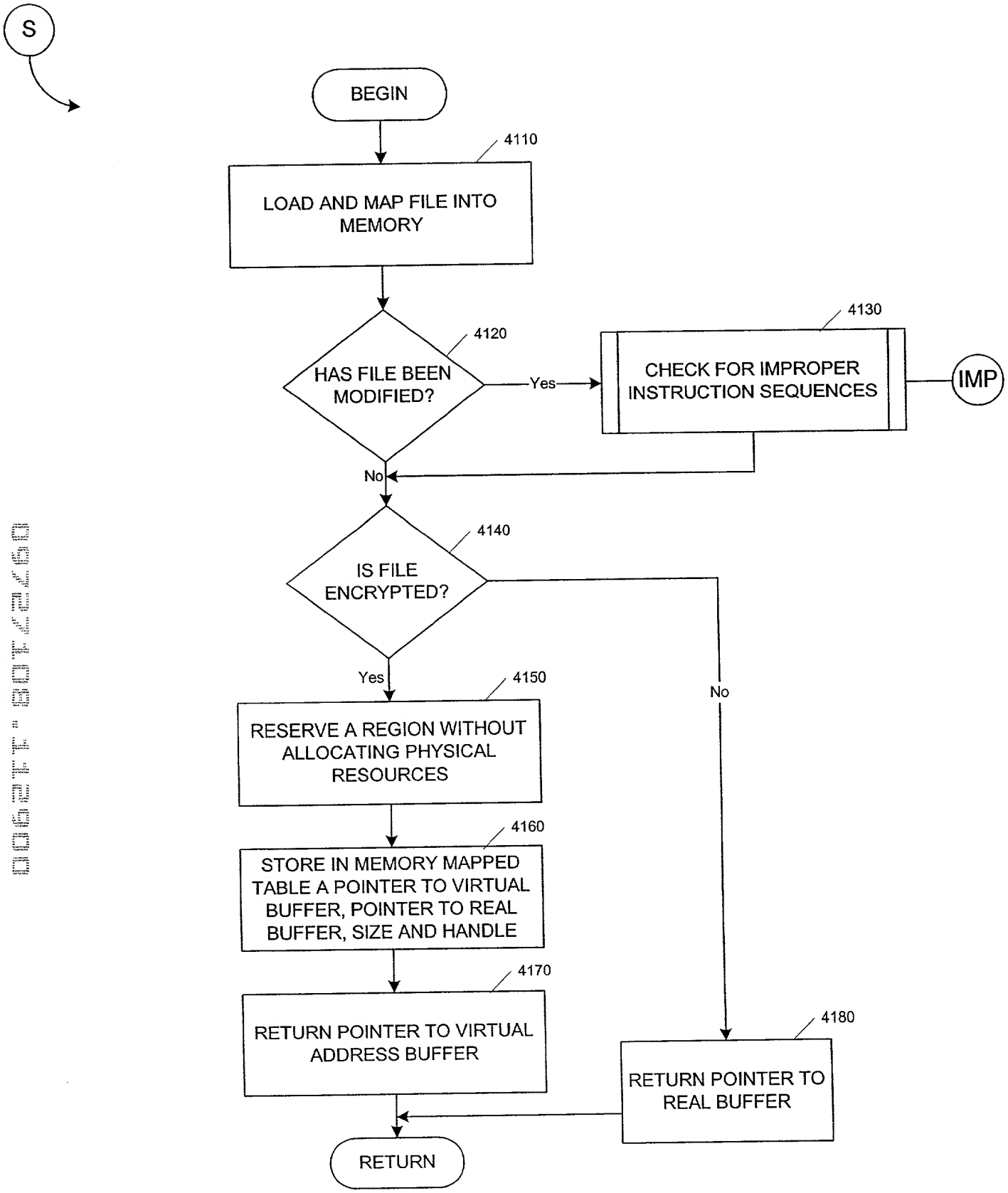
S
BEGIN
4110
LOAD AND MAP FILE INTO MEMORY
4120
HAS FILE BEEN MODIFIED?
Yes
4130
CHECK FOR IMPROPER INSTRUCTION SEQUENCES
IMP
No
4140
IS FILE ENCRYPTED?
Yes
No
4150
RESERVE A REGION WITHOUT ALLOCATING PHYSICAL RESOURCES
4160
STORE IN MEMORY MAPPED TABLE A POINTER TO VIRTUAL BUFFER, POINTER TO REAL BUFFER, SIZE AND HANDLE
4170
RETURN POINTER TO VIRTUAL ADDRESS BUFFER
4180
RETURN POINTER TO REAL BUFFER
RETURN

FIG. 41

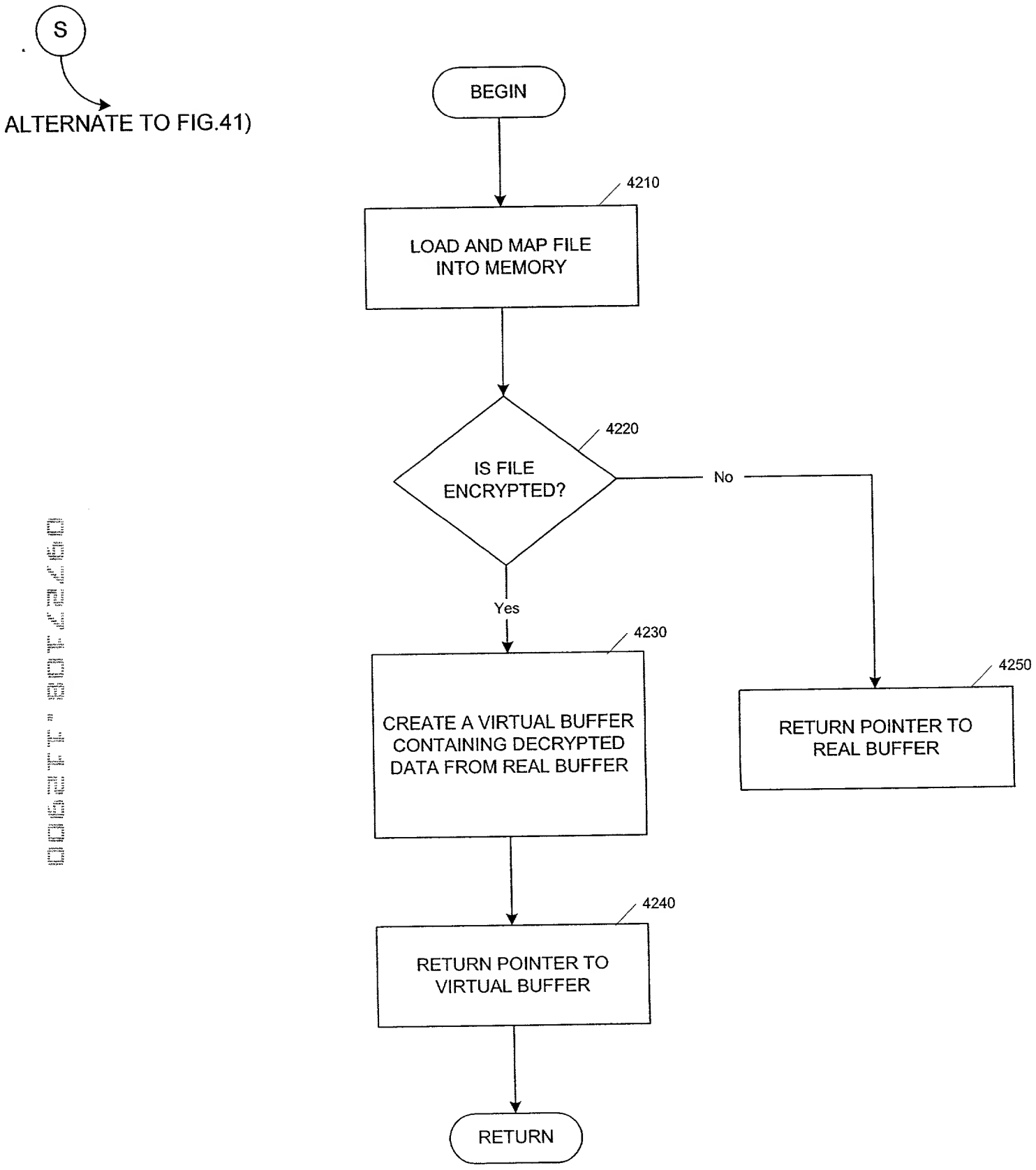
S
ALTERNATE TO FIG.41)
BEGIN
4210
LOAD AND MAP FILE INTO MEMORY
4220
IS FILE ENCRYPTED?
No
Yes
4230
CREATE A VIRTUAL BUFFER CONTAINING DECRYPTED DATA FROM REAL BUFFER
4250
RETURN POINTER TO REAL BUFFER
4240
RETURN POINTER TO VIRTUAL BUFFER
RETURN

FIG. 42

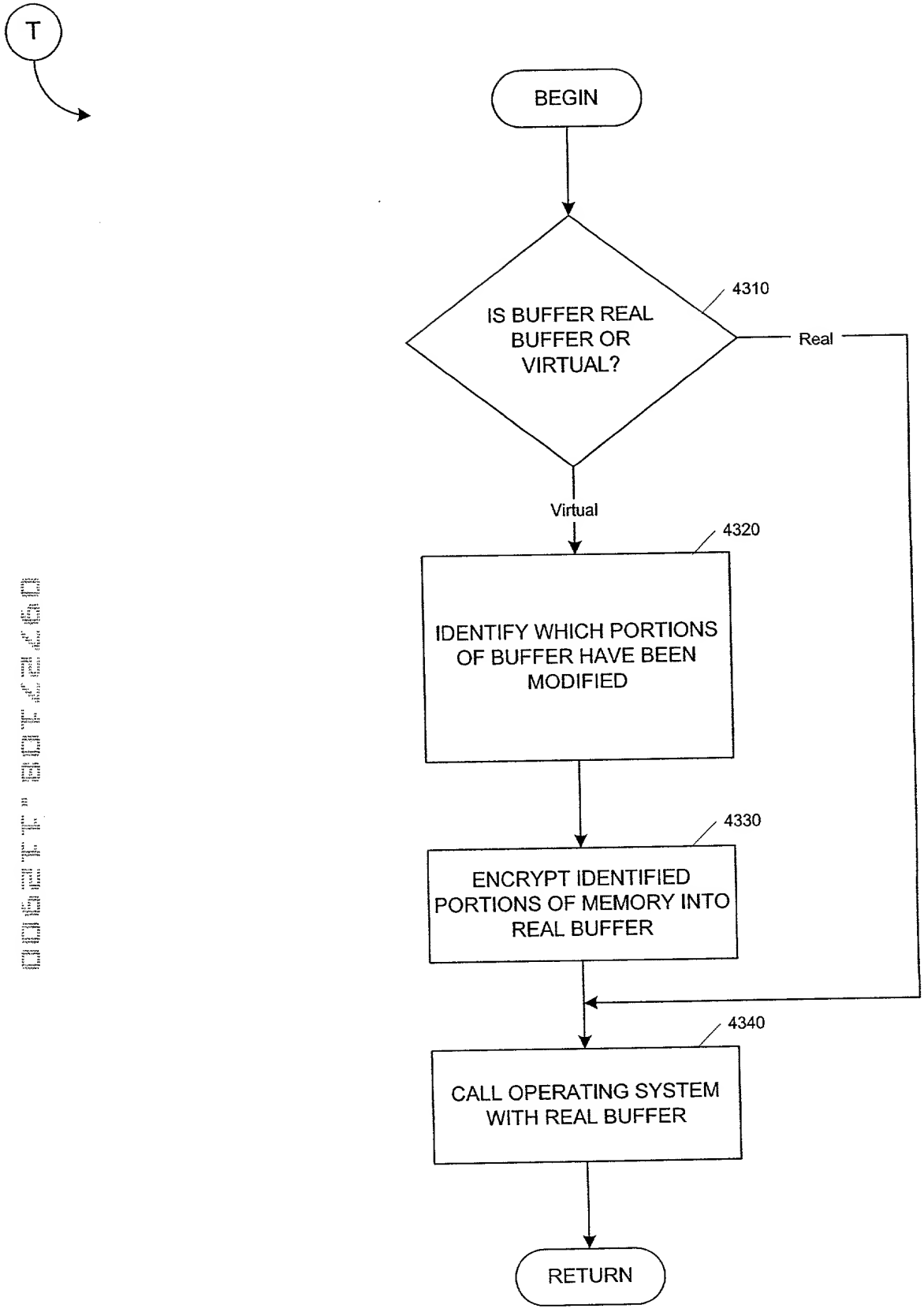
T
BEGIN
IS BUFFER REAL BUFFER OR VIRTUAL?
4310
Real
Virtual
4320
IDENTIFY WHICH PORTIONS OF BUFFER HAVE BEEN MODIFIED
4330
ENCRYPT IDENTIFIED PORTIONS OF MEMORY INTO REAL BUFFER
4340
CALL OPERATING SYSTEM WITH REAL BUFFER
RETURN

FIG. 43

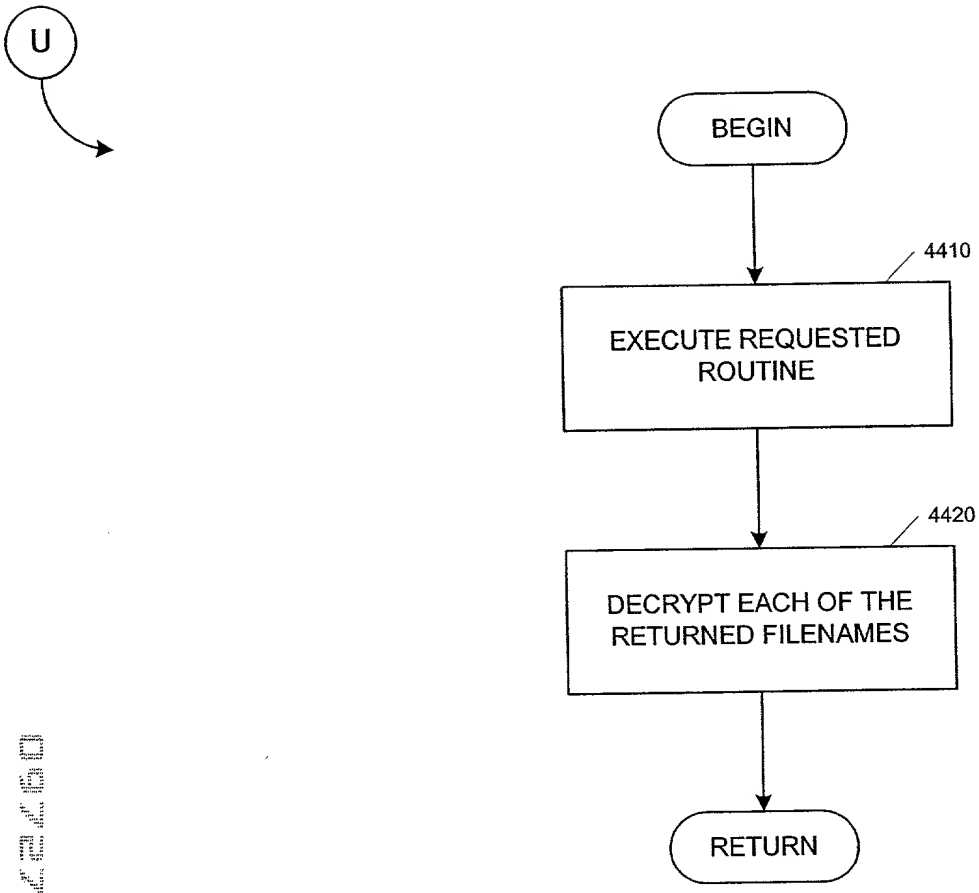
U
BEGIN
4410
EXECUTE REQUESTED ROUTINE
4420
DECRYPT EACH OF THE RETURNED FILENAMES
RETURN

FIG. 44

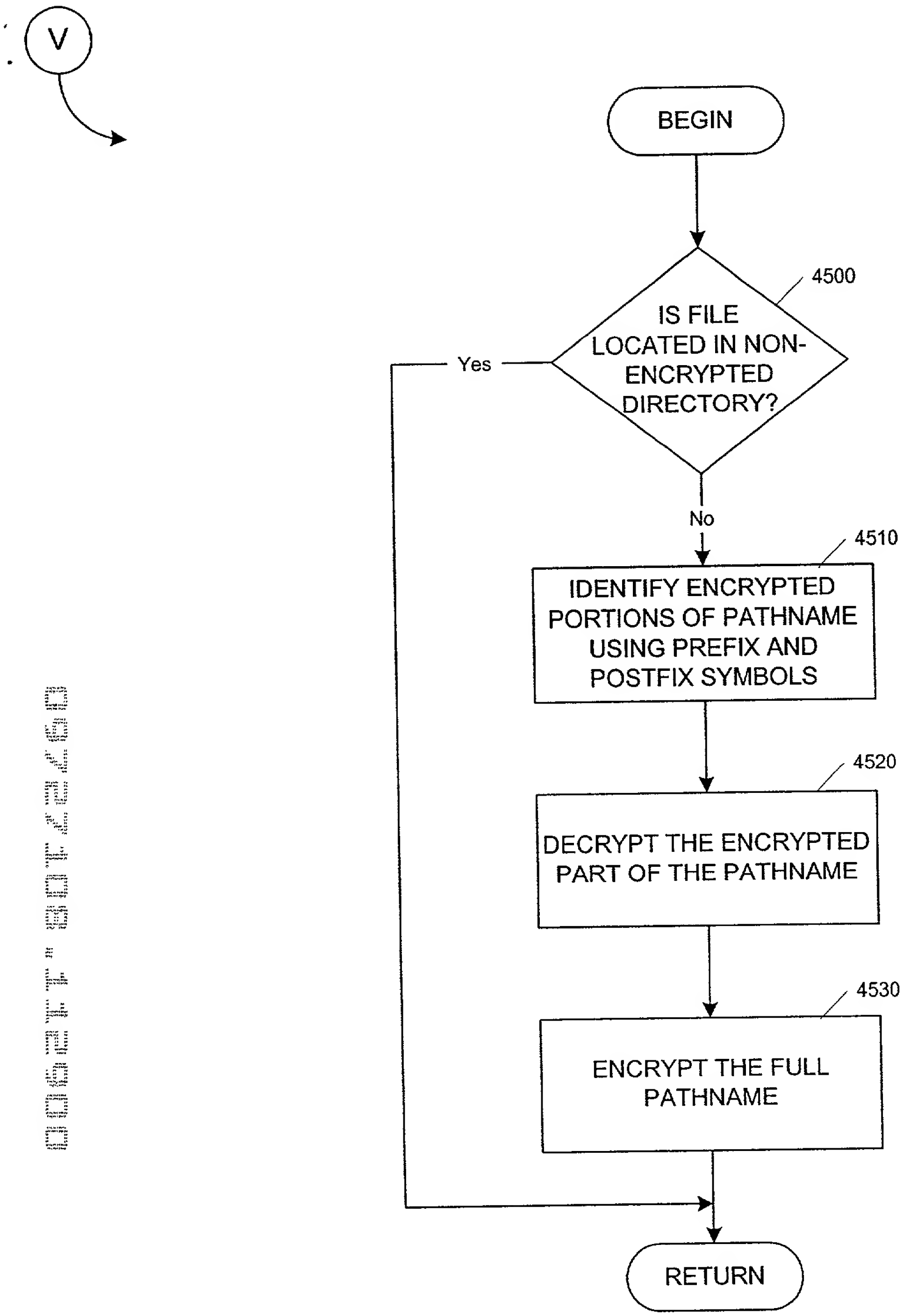
V
BEGIN
IS FILE LOCATED IN NON-ENCRYPTED DIRECTORY?
4500
Yes
No
IDENTIFY ENCRYPTED PORTIONS OF PATHNAME USING PREFIX AND POSTFIX SYMBOLS
4510
DECRYPT THE ENCRYPTED PART OF THE PATHNAME
4520
ENCRYPT THE FULL PATHNAME
4530
RETURN

FIG. 45

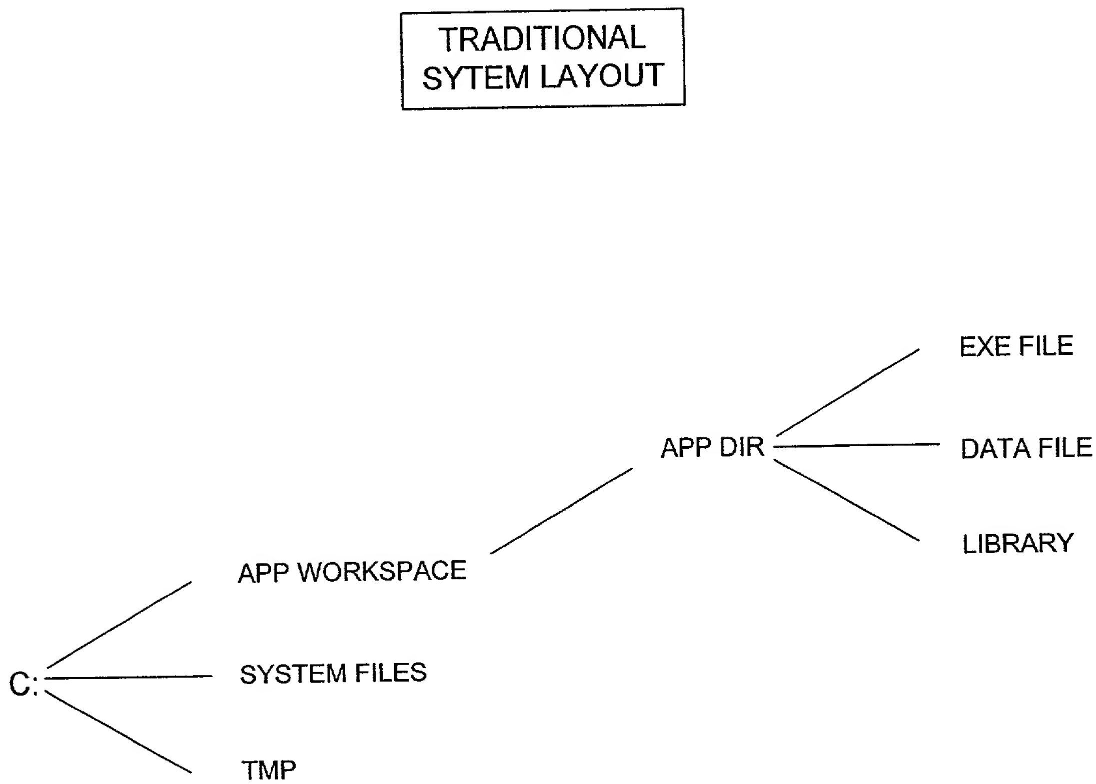
TRADITIONAL
SYTEM LAYOUT
EXE FILE
APP DIR
DATA FILE
LIBRARY
APP WORKSPACE
C:
SYSTEM FILES
TMP

FIG. 46

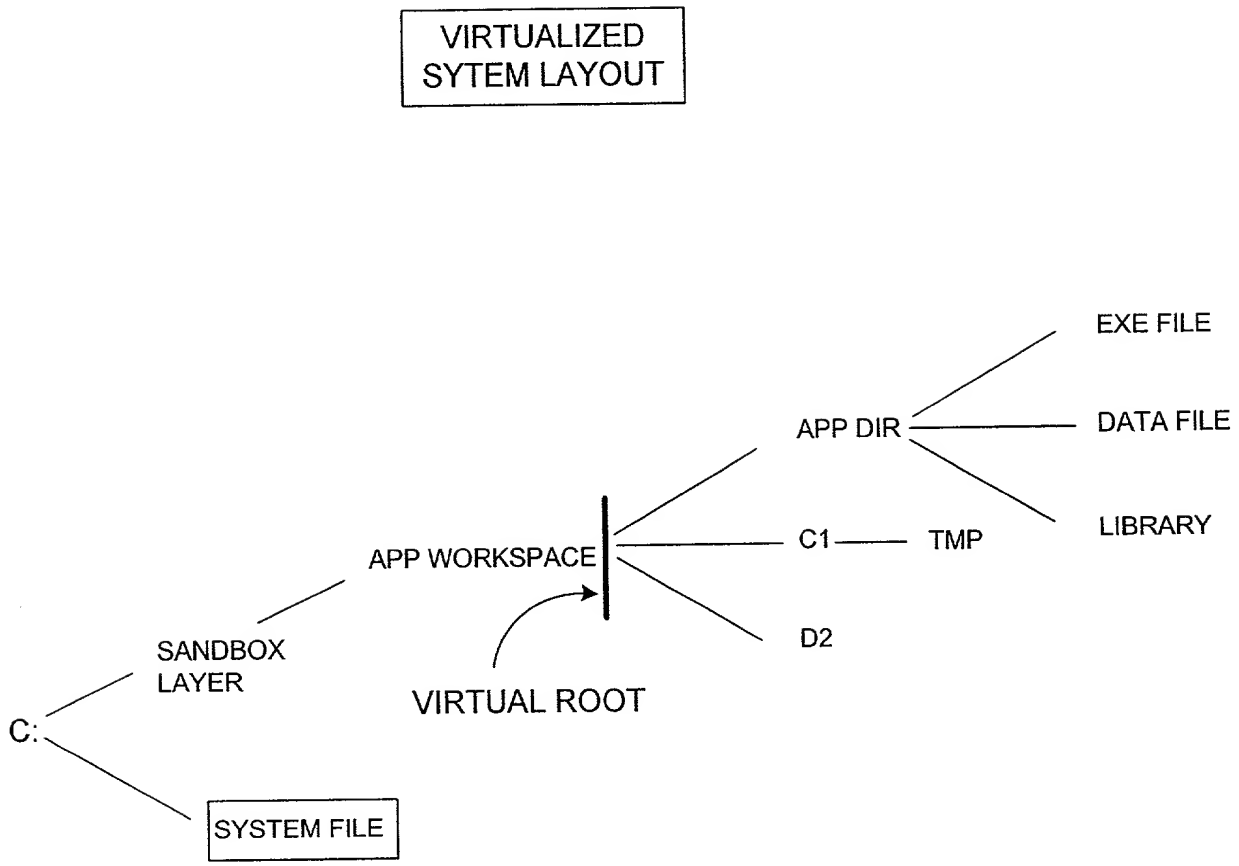

FIG. 47

SOCKET TABLE

4800

4804 "ENTRY" LOCAL SOCKET STRUCTURE	4812 REMOTE SOCKET STRUCTURE	4816 SOCKET STATUS	4820 SOCKET OPTIONS	4824 SEND QUEUE	4828 RECEIVE QUEUE	4832 CONNECTION QUEUE

SOCKET STRUCTURE

- UNIQUE SOCKET ID
- SOCKET TYPE
- PROTOCOLS
- OPTIONS
- NETWORK ADDRESSES
- EVENT
- FAMILY
- BLOCKING

SOCKET STATUS

- UNCONNECTED
- RECEIVING
- SENDING
- LISTENING
- CONNECTED
- DISCONNECTED
- TERMINATED
- SHUTDOWN
- BOUND
- CONNECTING

FIG. 48

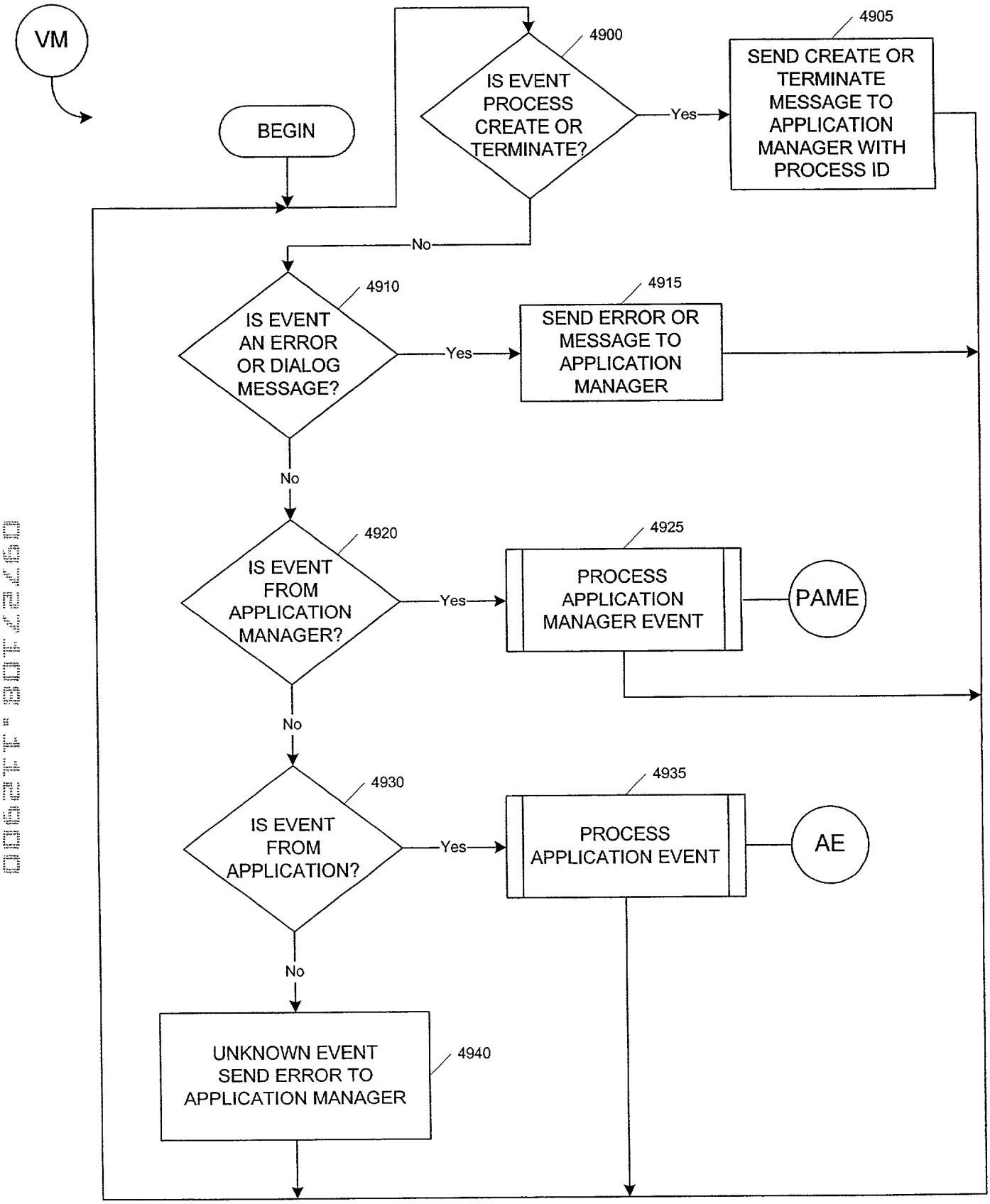
VM
BEGIN
4900
IS EVENT PROCESS CREATE OR TERMINATE?
Yes
4905
SEND CREATE OR TERMINATE MESSAGE TO APPLICATION MANAGER WITH PROCESS ID
No
4910
IS EVENT AN ERROR OR DIALOG MESSAGE?
Yes
4915
SEND ERROR OR MESSAGE TO APPLICATION MANAGER
No
4920
IS EVENT FROM APPLICATION MANAGER?
Yes
4925
PROCESS APPLICATION MANAGER EVENT
PAME
No
4930
IS EVENT FROM APPLICATION?
Yes
4935
PROCESS APPLICATION EVENT
AE
No
UNKNOWN EVENT SEND ERROR TO APPLICATION MANAGER
4940

FIG. 49

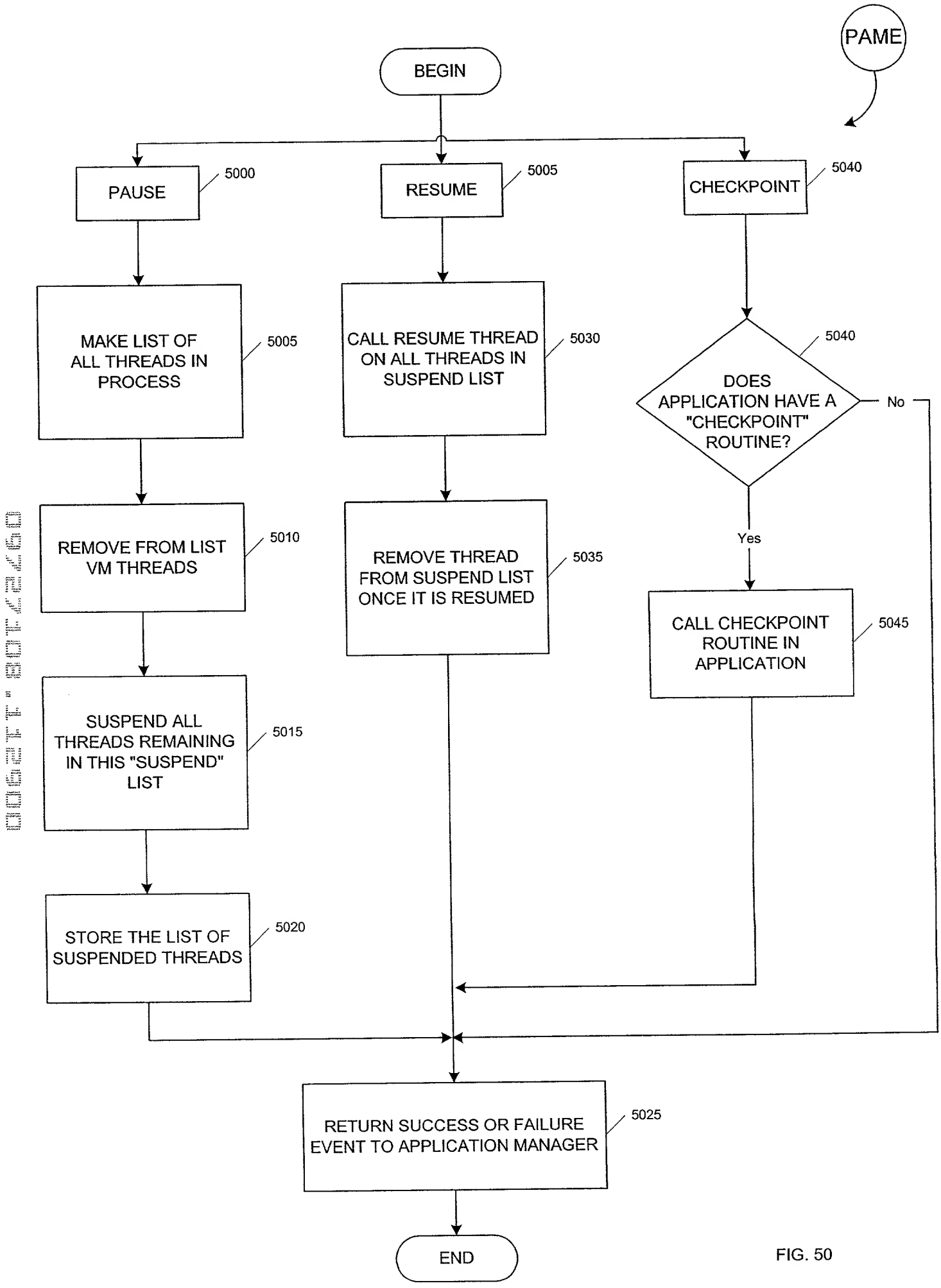
PAME
BEGIN
PAUSE
5000
RESUME
5005
CHECKPOINT
5040
MAKE LIST OF ALL THREADS IN PROCESS
5005
CALL RESUME THREAD ON ALL THREADS IN SUSPEND LIST
5030
DOES APPLICATION HAVE A "CHECKPOINT" ROUTINE?
5040
No
Yes
REMOVE FROM LIST VM THREADS
5010
REMOVE THREAD FROM SUSPEND LIST ONCE IT IS RESUMED
5035
CALL CHECKPOINT ROUTINE IN APPLICATION
5045
SUSPEND ALL THREADS REMAINING IN THIS "SUSPEND" LIST
5015
STORE THE LIST OF SUSPENDED THREADS
5020
RETURN SUCCESS OR FAILURE EVENT TO APPLICATION MANAGER
5025
END

FIG. 50

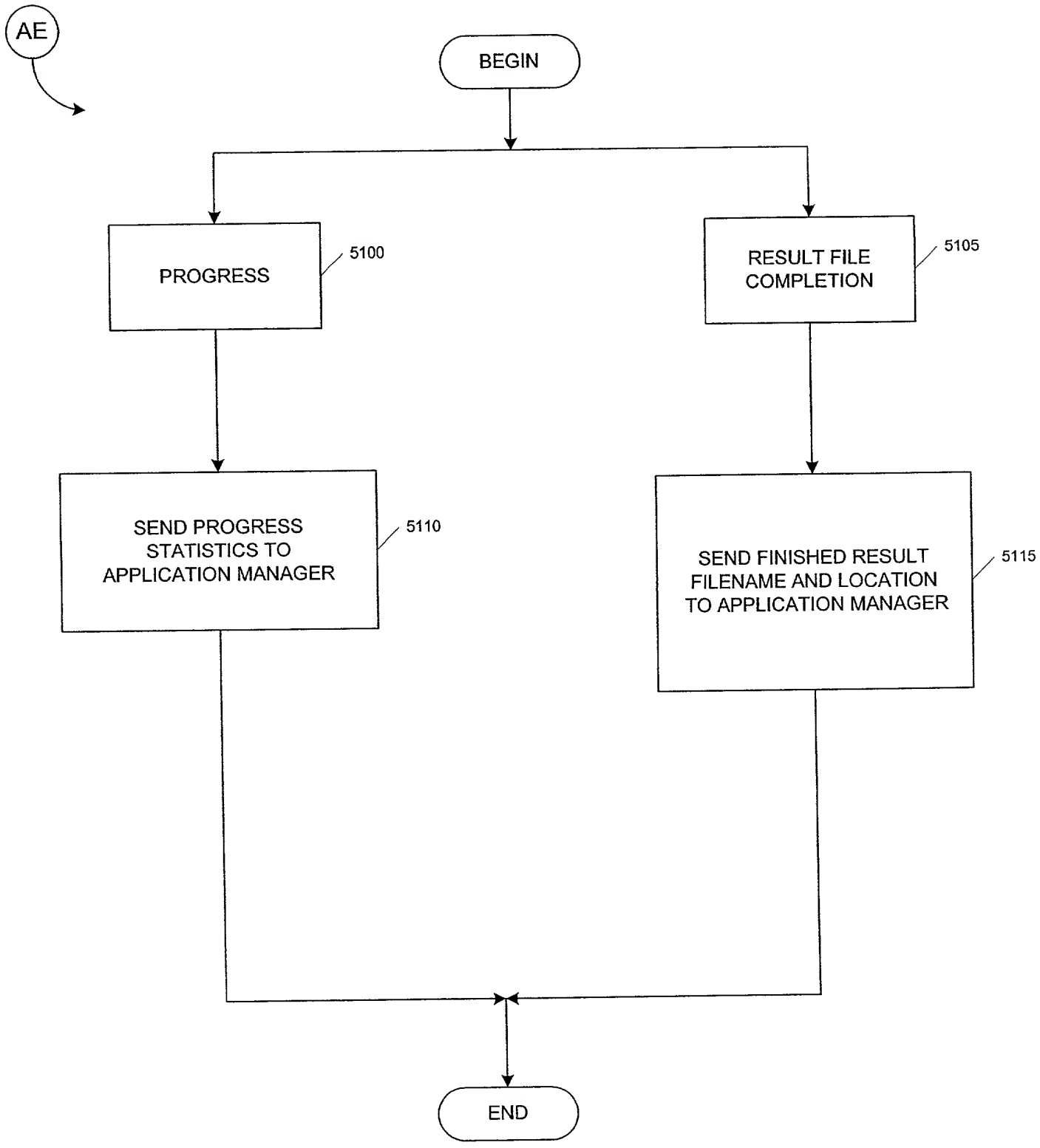
AE
BEGIN
PROGRESS
5100
RESULT FILE COMPLETION
5105
SEND PROGRESS STATISTICS TO APPLICATION MANAGER
5110
SEND FINISHED RESULT FILENAME AND LOCATION TO APPLICATION MANAGER
5115
END

FIG. 51